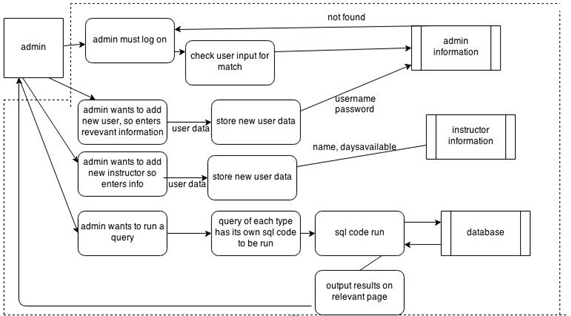
# Design

## Overall system design

|  |  |
| --- | --- |
| Input | Process |
| Customer details:  Firstname, surname, address ln 1, address ln 2,postcode, email, dob,contact no, emergency, password  Admin details:  Name, password  Group course info:  Courseno, clientid  Log on information(admin/user):  Id, password  Instructor info:  Instructor, dayavailable  Group course:  Coruseno, activity, instructor, skill, datetime, max entrants, duration, agerange,slope.  private course details:  courseno, clientid, Activity, instructor, skill, date, time, duration, age range, slope,  party course details:  courseno, clientid, activity, skill, numberadults, number infants, numberchildren, numberteens, , duration, date time, slope.  For booking onto group courses:  Id, courseno,  For each line of customergroup course:  Id, coursenumber | Look up all courses in the stated user criteria,  Combine user inputs into dates,  Read and write inputs to their assigned tables in the database  Query next user id for user to use as theirs  Print out reports:  Customer course lists  Course lists for the query  Individual instructor timetables  Delete and add records to and from a table  Run sql queries for validation purposes  Uses sql to determine the coursed based on a criteria  Determines the slope the course will be on based on a set of asp criteria  Sets the instructor based on the days they are available and assigns them on the table.  Updates various tables and records based on input values |
| Tables | Output |
| Table of admins  Table of course numbers,  Table of customers id’s assigned to grpcourses  Customer table  Group course information table  Table of instructors  Party table  Private table | Display: course list  Customer list  Customer id on add user page    Reports:  Customer course lists  Course lists for query  Individual instructor timetables |

### Customer DFD

This is a dfd to show what happens when the user visits the site, and is the main framework around the design of the project.

Admin staff DFD

This is the dfd for what happens when admin staff use the system, as you can see this is a small sample of all of the tasks an administrator can do, but the entire amount is discussed below.

### Client system flow chart:(shows the main system plan)D:\Users\toby\Desktop\Dropbox\Write up\design\welcomefd (1) (1).jpgAdmin system flow chart:(shows the main system plan)D:\Users\toby\Desktop\Dropbox\Write up\design\adminwelcomefd (1).jpg

## Data sources and destinations for new system

|  |  |  |
| --- | --- | --- |
| **Source of data** | **Method of data collection** | **Data items** |
| Customer | Website to create an account | First name, last-named, address line 1, address line 2, postcode, date of birth, emergency phone number, email, contact phone number, password |
| Instructor | Retrieving what time the instructor is available to teach | Time, day |
| Database | Retrieving customerid for a new customer | Customerid |
| Database | Retrieving list of courses for set day | Times, levels, instructors, clientid |
| Admin | Input by admin regarding a group course | Coruseno, activity, instructor, skill, datetime, max entrants, duration, agerange,slope. |
| Customer/admin | Log on on the site to book a course | id , password |
| Customer | Information on the course they wish to be booked on to. | Group course:Coruseno, activity, instructor, skill, datetime, max entrants, duration, agerange,slope.  private course details:courseno, clientid, Activity, instructor, skill, date, time, duration, age range, slope,  party course:courseno, clientid, activity, skill, numberadults, number infants, numberchildren, numberteens, , duration, date time, slope. |

|  |  |  |
| --- | --- | --- |
| **Destination of data** | **Method of data output** | **Data items** |
| Database | Input the user data into a new row under their individual clientid | First name, lastname, address line 1, address line 2, postcode, date of birth, emergency phone number, email, contact phone number, clientid, password |
| Instructor | List of the times and dates they will be teaching presented to them | Time, date, level, customers. |
| Admin | List of all courses being run during set day | Times, levels, instructors, coursed |
| Customer | Giving the customer a clientid |  |
| Customer | Displaying group courses that are NOT full, for them to book onto. | Group course:Coruseno, activity, instructor, skill, datetime, max entrants, duration, agerange,slope |
| Customer | Informing the customer if a course is full or not |  |
| Admin | Output from queries which they enter on a variety of query pages of which they input criteria to see who is on which course etc |  |
| Admin | Displaying all party courses available. | courseno, clientid, activity, skill, numberadults, number infants, numberchildren, numberteens duration, date time, slope. |
| Admin | Query to search the details of an individual user. | First name, lastname, address line 1, address line 2, postcode, date of birth, emergency phone number, email, contact phone number, password |
| Customer | Calculating their individual coursed for the course they are booking to, via asp | courseno |

## C:\Users\toby\Desktop\Dropbox\Write up\design\modularstructure.jpgDescription of modular structure of system

## Definition of data requirements (Design Data Dictionary)

Tbl customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tblcustomer** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| Clientid | Autonumber | 2 char | Must not be more than one under same id | 23 | |
| Firstname | text | 25 char | Must not be null | Toby | |
| Lastname | Text | 25 char | Must not be null | White | |
| Adressln1 | Text | 25 char | Must not be null | 5 acrefield drive | |
| Adressln2 | Text | 25 char | Must not be null | rawtenstall | |
| Postcode | Text | 9 char | Must not be null | Bb4 8du | |
| Email | Text | 50 char | Must not be null or invalid email address | Twhite\_96@live.co.uk | |
| Dob | date | 10 char | Must not be null  Must be a date  Must be valid date | 28/08/96 | |
| Phoneno | Text | 11 char | Must not be null | 07862852610 | |
| Emergency | text | 11 char | Must not be null | 01706213052 | |
| Password | Text | 20 char | Must not be null | tobywhite | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tblinstructor** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| Instructor | Text | 20 char | None | Dave | |
| dayavailable | Text | 10 char | None | wednesday | |
| activity | Text | 12 char | none | snowboarding | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tblparty** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| **COURSENO** | Autonumber | 2 char |  | 12 | |
| ClientID | number | 2 Char | Must not be null | 23 | |
| Activity | text | 12 char | Must not be null | Skiing | |
| Skill | Text | 12 char | Must not be null unless activity =tubing | Beginner | |
| numberadults | number | 1 Char | **Must be at least 1 combined** | 1 | |
| numberinfants | number | 1 Char | **Must be at least 1 combined** | 0 | |
| numberchildren | number | 1 Char | **Must be at least 1 combined** | 1 | |
| numberteens | number | 1 Char | **Must be at least 1 combined** | 1 | |
| Date | Date/time | 20 Char | Must not be null  Must be a date  Must be a valid date | 12/12/13 | |
| time | Date/time | 20 Char | Must not be null  Must be between 12-5 | 12/12/13 12:00 | |
| Duration | number | 1 Char | Must not be null | 1 | |
| Slope | Text | 6 Char | None | main | |
|  |  |  |  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tblgroup** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| **COURSENO** | Autonumber | 2 char |  | 12 | |
| Activity | text | 12 char | Must not be null | Skiing | |
| Instructor | Text | 16 Char | Must not be null | Dave | |
| Skill | Text | 12 char | Must not be null unless activity =tubing | Beginner | |
| date | Date/time | 20 Char | Must not be null  Must be a date  Must be a valid date | 12/12/13 | |
| time | Date/time | 20 Char | Must not be null  Must be between 12-5 | 12/12/13 12:00 | |
| Duration | number | 1 Char | Must not be null | 1 | |
| maxentrants | Number | 2 Char | Must not be null | 7 | |
| agerange | Agerange | 5 Char | Must not be null | 2-5 | |
| Slope | Text | 6 Char | None | main | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tblprivate** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| **COURSENO** | Autonumber | 2 char |  | 12 | |
| ClientID | number | 2 Char | Must not be null | 23 | |
| Activity | text | 12 char | Must not be null | Skiing | |
| Instructor | Text | 16 Char | Must not be null | Dave | |
| Skill | Text | 12 char | Must not be null unless activity =tubing | Beginner | |
| date | Date/time | 20 Char | Must not be null  Must be a date  Must be a valid date | 12/12/13 | |
| time | Date/time | 20 Char | Must not be null  Must be between 12-5 | 12/12/13 12:00 | |
| Duration | number | 1 Char | Must not be null | 1 | |
| agerange | Agerange | 5 Char | Must not be null | 2-5 | |
| Slope | Text | 6 Char | None | Main | |
|  |  |  |  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tblcustgroup** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| **COURSENO** | Autonumber | 2 char |  | 12 | |
| **Clientid** | Autonumber | 2 char | Must not be more than one under same id | 23 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TblAdmin** | | | | |
| **Name** | **Data type** | **format** | **Validation** | **Example** | |
| Adminname | Text | 10 char | None | Toby | |
| Password | Text | 16 char | None | password | |

## Description of record structure (if appropriate)

TBLCUSTOMER(ClientID, lastname, firstname, addressln1, adressln2, postcode, dob, phoneno, emergency, email, password)

TBLINSTRUCTOR(instructor, activity,dayavailable)

TBLPARTY(Courseno, ClientID, activity, skill, numberadults, numberinfants, numberchildren, numberteens, date,time, duration, slope)

TBLGROUP(COURSENO, activity, instructor,skill, date,time, maxentrants,duration, agerange, slope)

TBLPRIVATE(Courseno,clientID,activity, instructor, skill, date,time, duration, agerange, slope)

TBLCUSTGROUP(COURSENO,ClientID)

TBLADMIN(adminname,password)

## Entity Relationship Diagram

Party course

Private course

Instructor

customer

customerGroup

Group course

## File organisation and processing (if appropriate) or database design including normalised relations (if appropriate)

The only relationship is linking the TBLGROUP(courseno, activity, instructor,skill, datetime, maxentrants,duration, agerange, slope) and TBLCUSTGROUP(COURSENO,ClientID)

Assigning one client to a group course which has many clients on it.

## sample of planned SQL queries (if appropriate)

I plan to use a wide variety of SQL queries as possible, including deleting and adding records to a table, using sql for validation purposes to see if courses are fully booked and if so not to display them, and also various sql for queries made by the client and the admins, I feel that I will use a lot of sql, and will make a list of all of my sql queries on a latter page. I also think I will add code to update the information on the table, eg for admin staff to update the weeks instructors for course assignment, and for users to update their details themselves, on data entry forms from the user eg booking pages, SQL code will be used to add the contents to the table on the database, for the user logging in , or the majority of the queries the admin staff use to search for results of a criteria, will use sql SELECT function or SELECT DISTINCT to retrieve data from the database, on parts such as updating the instructors for the week ahead, I plan to use sql code for this as well, such as UPDATE function , which will update the instructors and the day they are available. Every field used that the user inputs in some way as well as some that are processed by the computer to get the value, will be used in sql coding at some point. There is a vast amount of sql code some of which have results which then become attributes of another sql sequence, for example checking if an instructor is free at a set time and if they are to return the user to the previous page. All of this combined will help to make the system fully efficient everything can run smoothly

## Identification of storage media

I plan to replicate the system I have at school between the servers the club uses and their main desk for their computers, with this new scheme, I have also told them that now they will be introducing electronic data, that they will have to back up all data on an external hard drive at the club and store it in a secure location, the person who developed the old website is good with computers, and has assured me he is fully capable of doing this. Server side script will of course be stored remotely on the relevant external server, however I have contacted the server company (ISSL ROSSENDALE) and they have assured me they regularly back up all data as part of the package the club already has, furthermore, master copies of the code will remain on my 3 external hard drives at home, so there is virtually no risk of data being lost. All pages will be stored on the existing web servers the club currently used, and all that is required is to move all web pages to a pen drive, and then place them on the web server and publish them. These pages will however be combined with the current website the club already has, hence there will be a ver similar look to the site as before, so the person in charge of the old site can simply directly add these pages on to the new server, along side the ones they currently have.

## Identification of suitable algorithms for data transformation, pseudocode of these algorithms

Set Instructor day equal to weekday name of date user input

If activity=snowboarding then

Set slope equals fun park

Else if skill is beginner then set slope equal to beginner

Else set slope to main slope

End if

Connect database

Set Instructor to result of table instructor where day available=instructor day and activity=activity

Set Course avail to get from tblprivate where instructor=instructor and date=date and time=time;

If coursavail is null then

Redirect to previous page

Else write to tblpriv

End if

Function get course number

Connect db

Set Lastrec to top 1(courseno)from tblprivate order by courseno desc

Write courseno is:(lastrec+1)

connect db

get all FROM tblprivate WHERE date>date1 And Date<date

do until EndOF

write table

for each value in Field of table

Write name

Write value

Finish table

loop

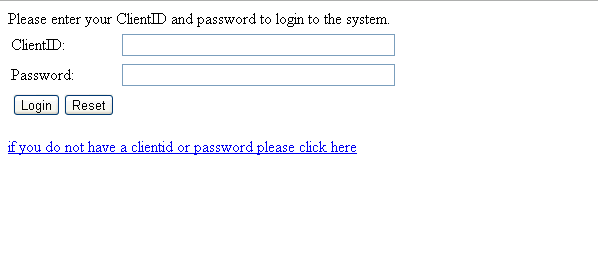
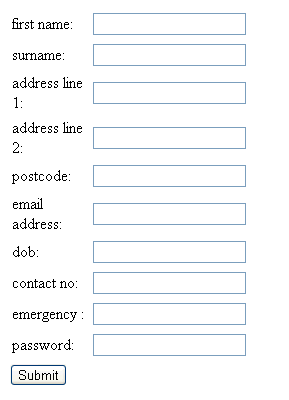
## UI sample of planned data capture and entry designs

When designing the site, I have done some prototyping as I feel this is the best way for the user to approve or offer improvements to the designs before the system is created, this allows the user to evaluate the designs, making the improvements offered as accurate and precise as possible. From past experience making prototypes I feel is the best method of design phase.

**Client first time logon**



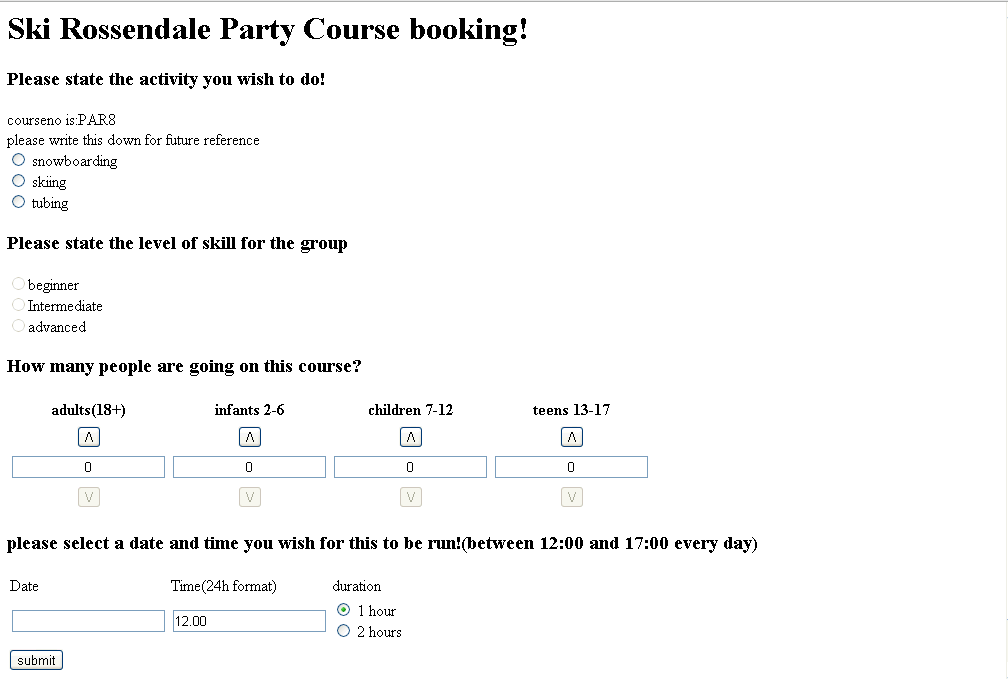
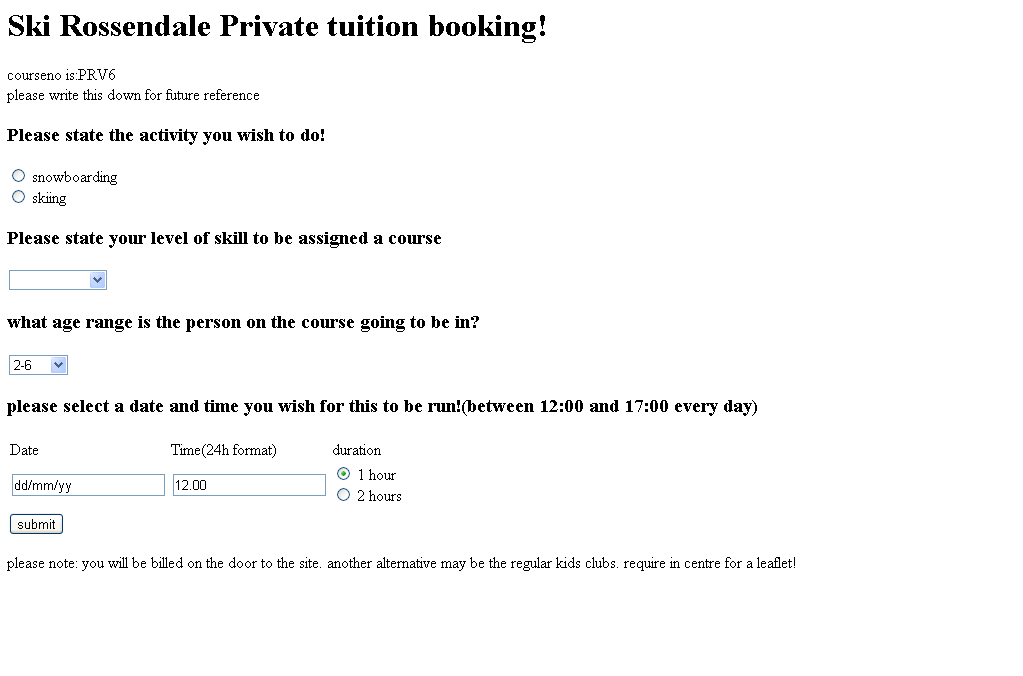
This is the page where a user will input their details for the first time, on submit it will be validated and sent to the table, all data on this will be input by the user, at the top however there will be an asp query from the database to show the next client id available(which will be the current clients id)



Once the user has registered, they will be directed here to log on

Following each log on, they will be given the opportunity to change their details if they wish, they can do that here.

**Course booking pages**



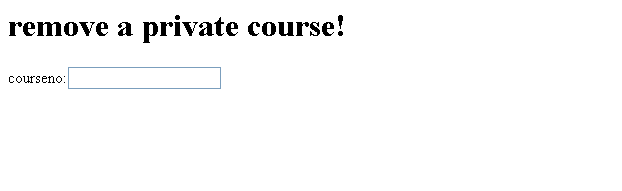
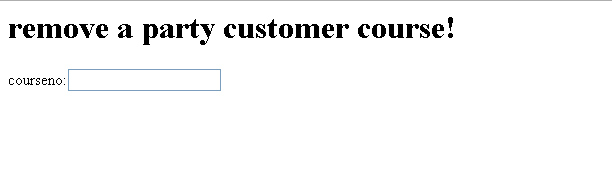
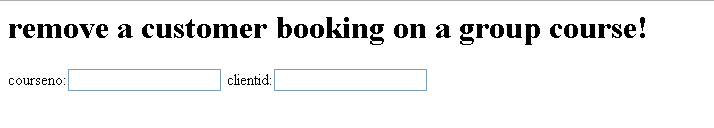
The user can then book one of three courses, private, group or party.

Here is the private booking page, this is for the user only, and for singular one to one lessons. As you can see there are two combo boxes, these are populated with values that are already assigned in the table, using select distinct SQL code, I have populated these with table values

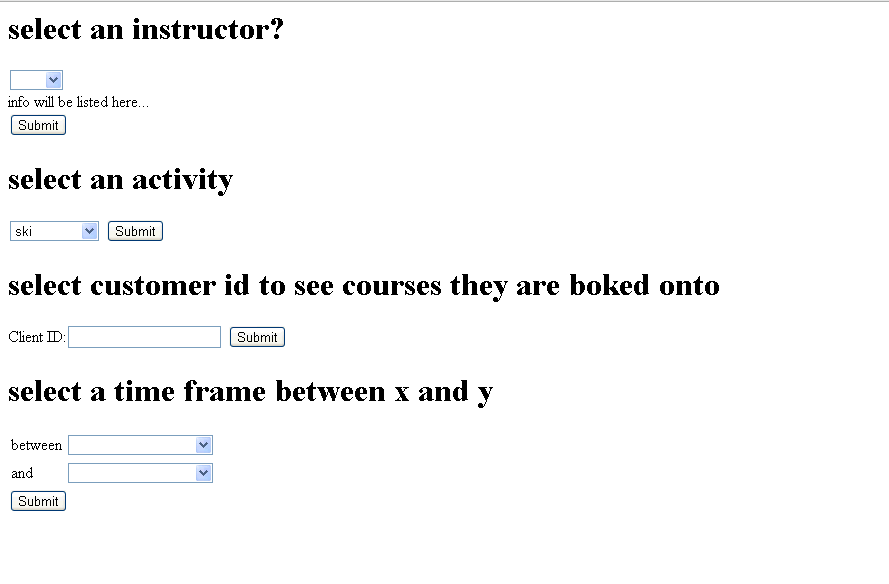
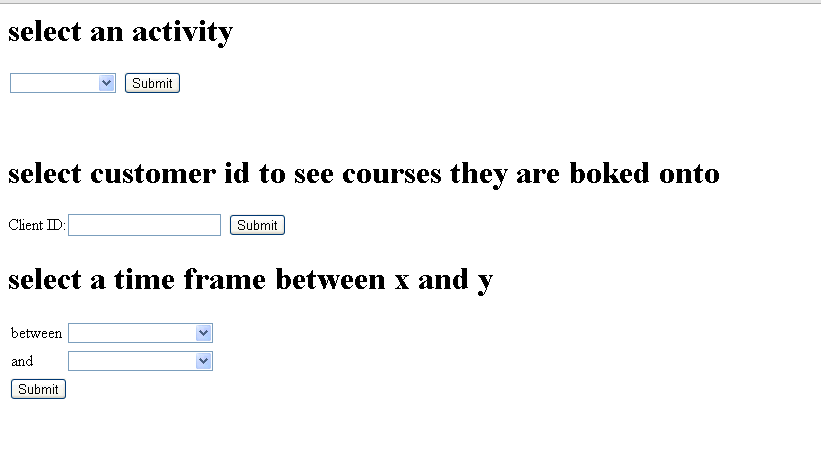
As you can see at the top of each page is the courseno: this will be populated via sql again using the select top 1 function to get the coursenumber of the next course to be assigned, it is also given a pretext so staff members can easily find which area to look in during queries

This page is very similar, all form attributes have been selected in accordance to the simplest way of the user entering their data, eg it is much more sensible to use a drop down box for skill level as opposed to them typing their skill level, and this helps to reduce system errors

***Admin***



This is a sample of the pages containing admin tools, mainly linking to sql commands linked to the database, I plan to make all these separate and have a main admin page linking to them all.

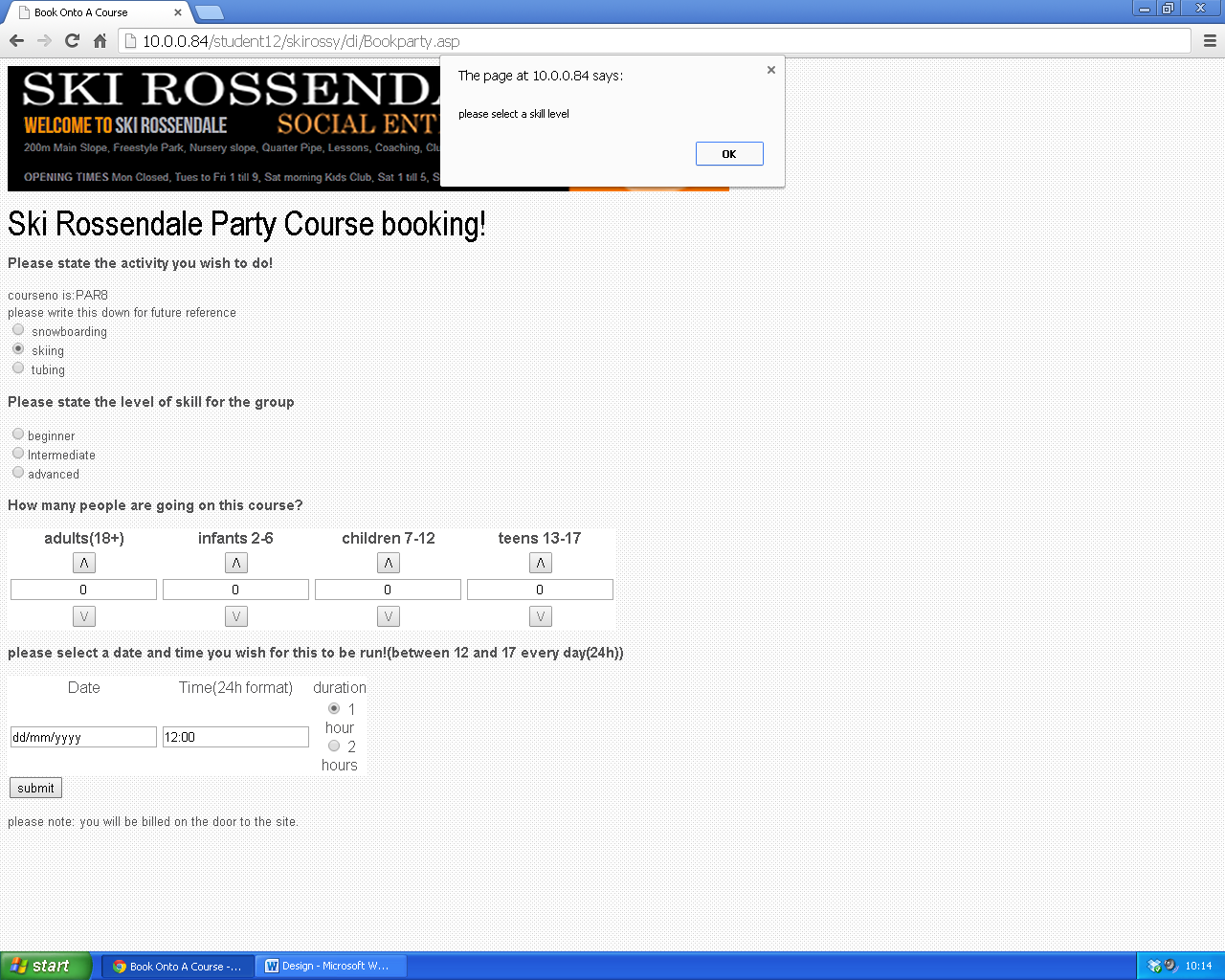


This is a sample of the query pages for admin staff, so they can search for courses based on the various criteria on the page, all combo boxes are populated via the SELECT DISTINCT function from the table, so the admin staff can get results based on almost any query they want.

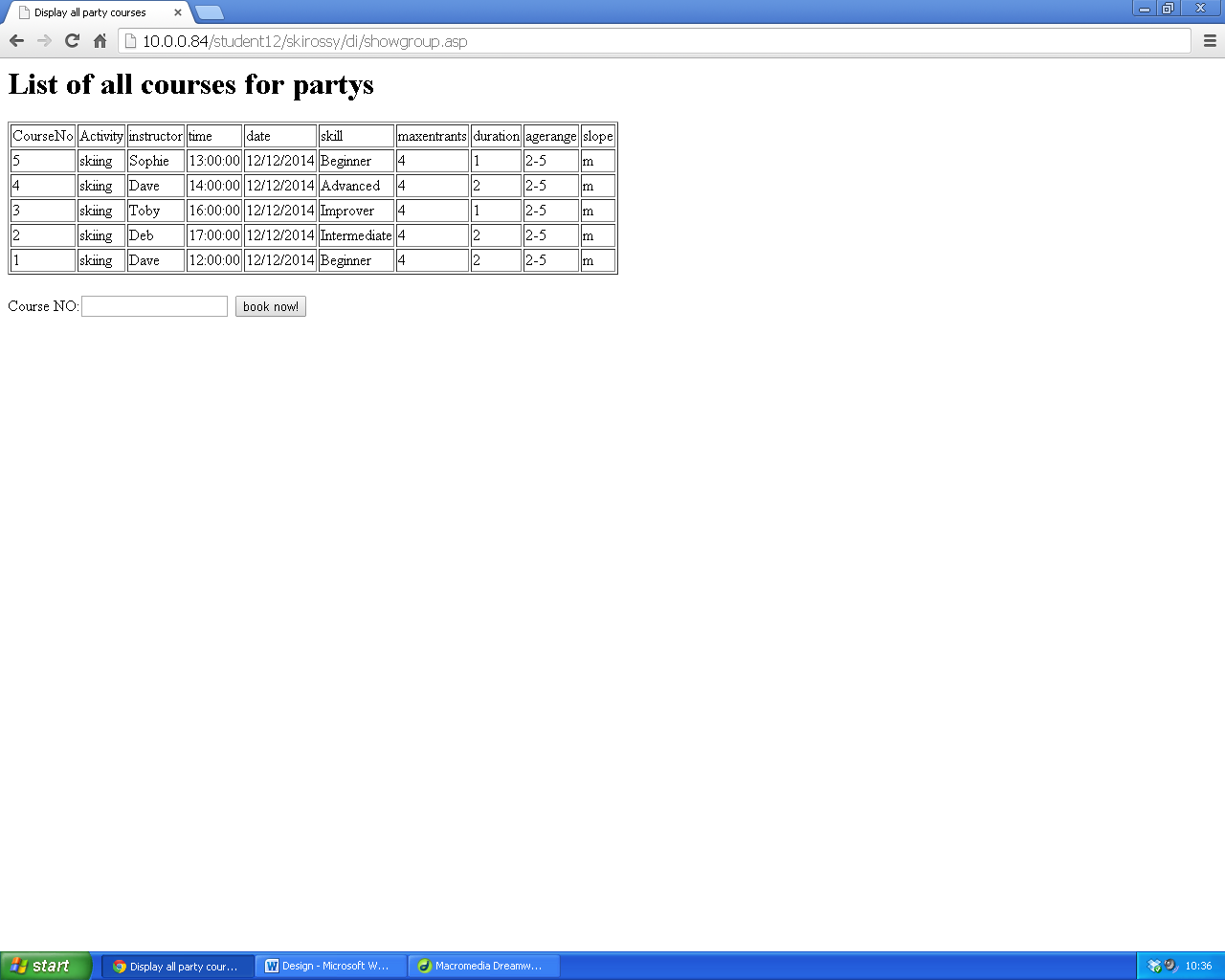
These queries have been placed on separate forms on the same page, as admin staff often are rushed and want to find things straight away; this helps them as all possible things are clearly accessible on one page when they need to run a query.

## UI sample of planned valid output designs

Naturally there is a large amount of JavaScript validation on the pages, when an erroneous data entry is given or data is not entered , the javscipt will alert the user and prevent them from continuing.

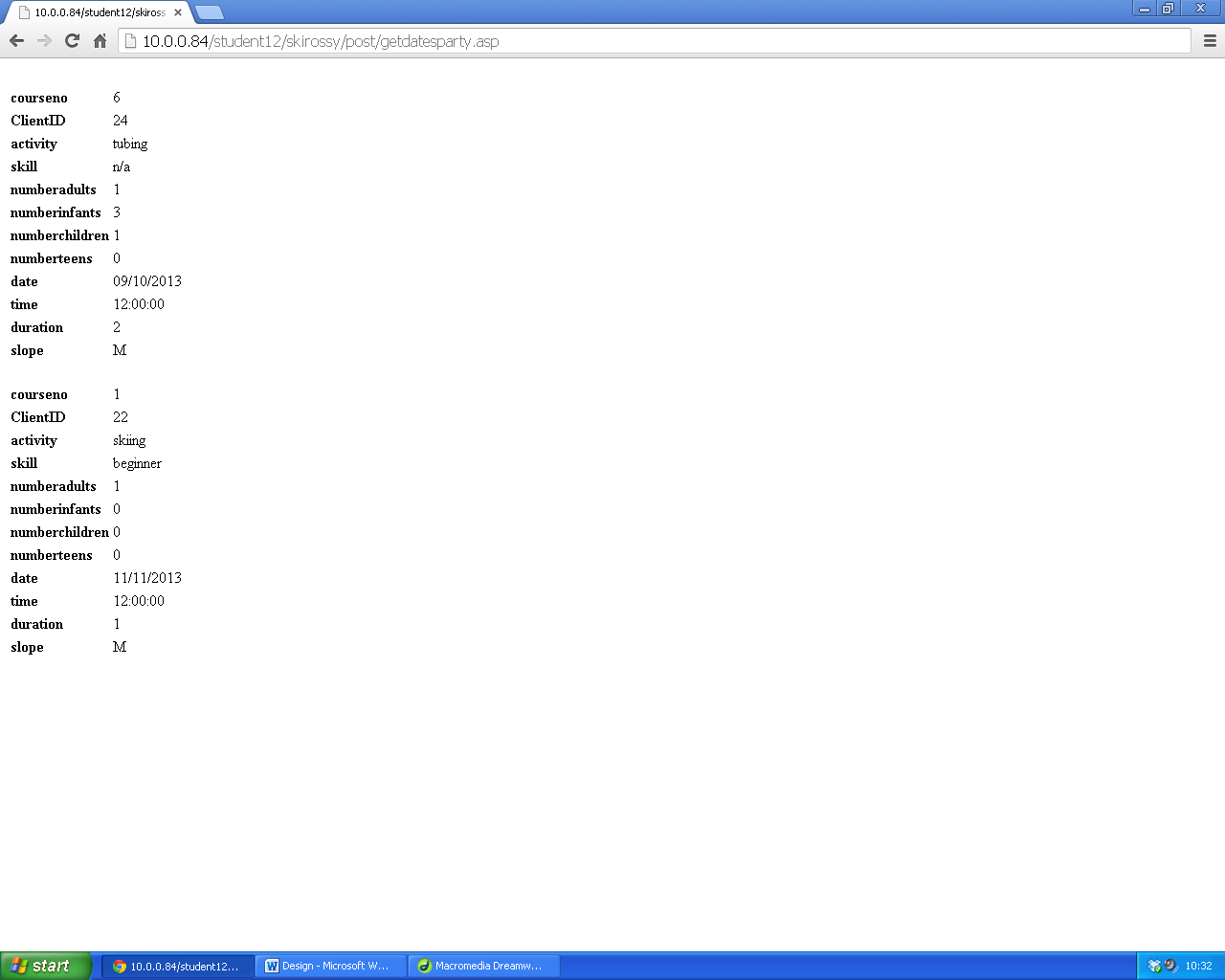


When booking a group course the user will select the activity list they want to see, for example skiing, so on the following page, SQL will find entries in the database of group courses that are not full, allowing the user to input the course number they would like to book onto



An example of an admin output is when the admin will use a query to see courses between a set of two dates and will then be displayed in a table, this will enable the user to have clearly displayed all courses over the set period in order to print it off etc, there will be sql to get the data then it will be written to a table.

For example if it is a party query,



## Description of measures planned for security and integrity of data

The system will have a two tier log on system, in order to gain any access to the site, users will have to register themselves and also choose a password to give them access to the site, this means all pages can only be seen by people who are registered, to save recursion of data, and prevent multiple bookings, there is then a level of which the admin staff can log onto to perform tasks such as edit , add, or remove entries from the database to name a few, again admins must choose a password before being able to access the admin area. This I feel makes the security of data much more secure as if users do anything they will leave an identity mark allowing whatever they do to be traced on the server.

## Description of measures planned for system security

The computer system will be secured in the office which requires a passcode to enter, and the server will be stored in a locked server room at the host’s warehouse. This means the system is very secure as data accessed in the office is mainly remote access to the server etc. as discussed above there are passwords required to gain access to the relevant levels of control through the site.

## Overall test strategy

I plan to test each section of my coursework the entire way through its implementation, with each new feature I add, I intend to test each individual criteria (where applicable) with one below the limit, on the limit, and one above the limit, alternatively I will input typical, extreme and erroneous values (where applicable)to the testing to get the best and most effective site available to the customer.

The majority of my validation code will be placed in an external JavaScript file containing all of the validation processes for user entry forms; I will use “response. Write” statements to test the majority of the validation code to see if it is running and carrying the correct variables, this will be done for each separate process on the JavaScript file, thus covering the majority of my validation testing.

My next set of code to be checked will be my sql statements on the asp pages, which will execute various commands such as writing to, deleting records from, and reading records in a database table, for these sql statements I intend to run the page as normal with values I have chosen above.

I will then need to test the log on pages that I have made, alongside the include files referring back to this on all pages which require the user to be logged on to view these pages, I must check that each page, given I have 2 different types of log on page (admin and user) I will need to check that their separate include files are connected to the correct pages and redirect correctly if the user is not logged in.

Finally I will need to do some integration testing, this mainly involves the passing of variables between asp pages using the post method, I will need to check these variables are passed correctly and do not become corrupted, as this is the main task behind my project, alongside this, I will also need to check that my JavaScript page(referred above) is also correctly retrieving values from each page correctly.

|  |  |
| --- | --- |
| Page name | Testing method |
| Data entry forms:  Addgroupcourse.asp  Addnewadmin.asp  Bookgroup.asp  Bookparty.asp  Bookpriv.asp  Logonreal.asp  Adminlogon.asp  Logon.asp  Searchcustomerdetails.asp  Inputcustomer.asp  Showgroup.asp- also displays results through sql command | I will test all of these forms with erroneous, borderline and normal values, the javascript should cover the majority of the validation, luckily, all validation is on one external javascript file, meaning it is a lot easier to organise this section. |
| Admin log on pages/user log on pages (ASP):  Admininclude.asp  Adminlogon1.asp  Include.asp  Logonreal1.asp | I will simply check these pages work with real, and false values |
| SQL command pages  Displaypartycourses.asp  Getactivitypriv.asp  Getactivityparty.asp  Getcustomers.asp  Getdatesparty.asp  Getdatespriv.asp  Getinstructorspriv.asp  Getpartycustomers.asp  Getprivcustomers.asp  Removecust.asp  Removecustgroupuser.asp  Removegrpcourse.asp  Removepartycustomer.asp  Removeprivcourse.asp  Searchprivcourse.asp  Sendadimn.asp  Sendgroup.asp  Sendparty.asp  Sendpriv.asp  Storeinuputcustomer.asp  Storenewgroup.asp | I will simply check these pages work with real, and false values, I will also check where applicable that each sql page runs the sql correctly with true, erroneous and extreme values to test this. |
| Asp query pages(admins) Data entry forms:  Searchprivcourse.asp  Searchparty.asp | I will test all of these forms with erroneous, borderline and normal values, the javascript should cover the majority of the validation, luckily, all validation is on one external javascript file, meaning it is a lot easier to organise this section. |
| Pages that do not require testing:  Csssheet.css  Logo.png |  |
| Validation pages:  Scripts.js | I will do the majority of tesing file using this page, I will test all functions with a variety of different values ,which I will detail in the test plan. |