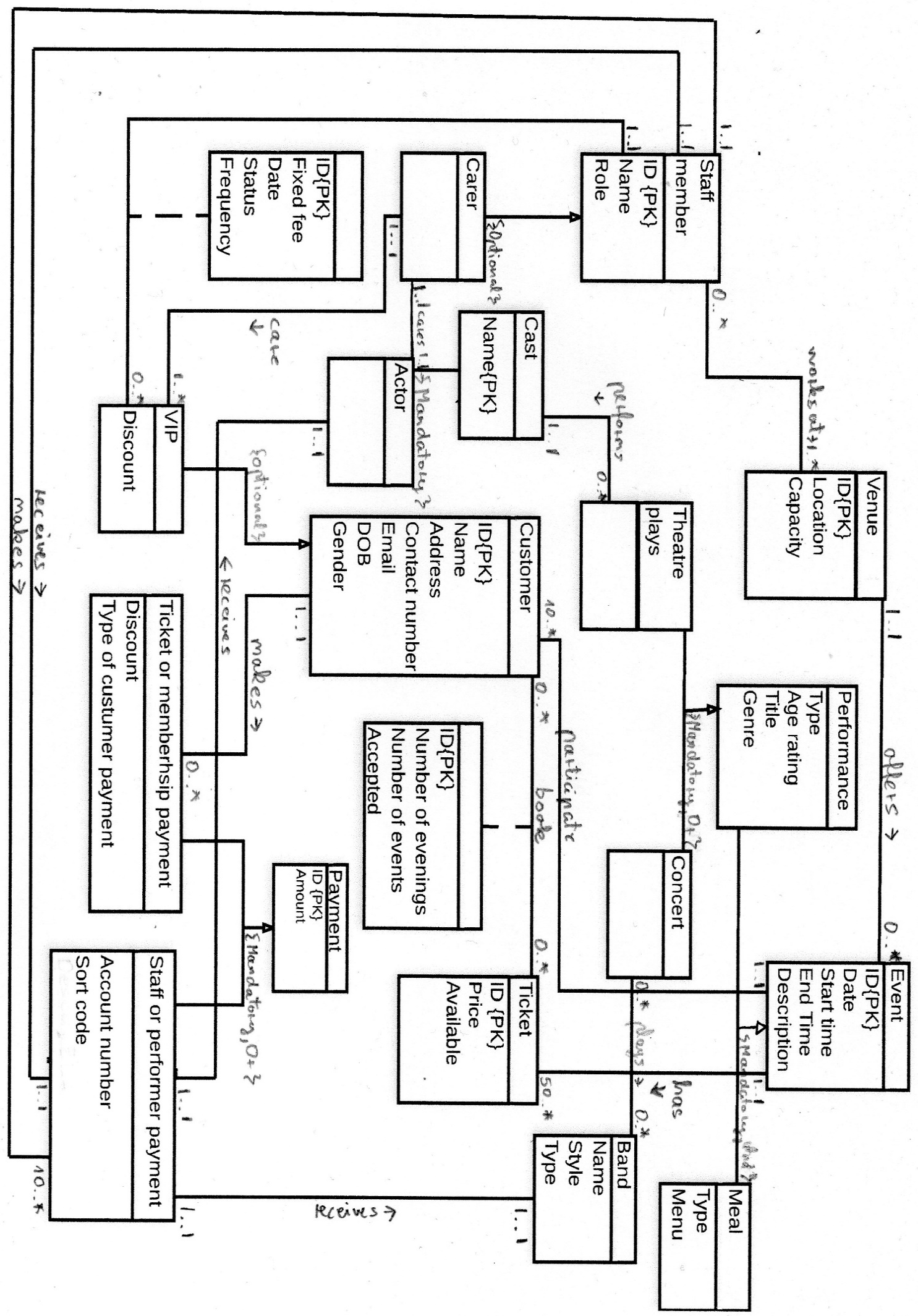
**Meirunas Smitas, Tuesday 11:00 am to 1:00 pm in Clipstone, G.102.**

**Part A**

**1.**

**1111.11+1**

**2.** Provide an executive summary of how you identified the **entity types** for PerfoSpace (including eventual generalised/specialised entities). In other words, explain and justify how you identified these entities in a written narrative in the first person. This needs to fit on no more than one page of the report.

I did read assignment and applied method for search nouns and noun phrases which are entities. Firstly, when I did read specification I highlighted nouns that makes easier to find entities. Staff member has to be included because company cannot exist without employees and in the text passage” employs a large number of staff”. Next included entity is Carer because cares about actor’s needs figured out from the phrase “upon request, some actors, perhaps, lead or high profile performers, can be assigned a particular member of staff to look after”. I identified Cast from the following passage “all the members of the cast of a play need to be registered”. Actor performs in theatre play so they are included in the conceptual diagram. Venue means the building of event. I considered the Event entity important because description mentions that events are organised by the company and include performances and meals. For the performance there are two types theatre plays and concerts. Theatre play and Concert are appearing to be entities as PerfoSpace has to host that. Next Band is entity because it can contain details of the bands evidence” The details of the bands”. The entity Meal is also important for the company database because Meals noun are available before or after a performance so customers can choose have meal or not with option of foods. Customers is another entity that was identified from the brief saying” customers need to register” this means company has customers. Tickets noun is mentioned a few times for example” sufficient tickets available” so this should be included in entities and in common sense events cannot exist without tickets. This VIP entity derives from Customer and provides important information about VIP customers. The brief mentions five types payments ticket, memberships, staff member payment, actor payment and bank payment. I decided combine the first two in one entity call Ticket or membership payment dealing with customers and last three in another entity call Staff or performer payment having to do staff members and performers. I created Payment generalisation entity including common attributes.

**3.** Provide an executive summary of how you identified the main **relationship types** for PerfoSpace. In other words, explain and justify how you identified these relationships in a written narrative.

I identified relationships between entities relating to each other. Staff member has relationship receives with Staff or performer payment as in text “employs a large number of staff” which means staff receives payments and performers too. Role as each member has it role as employee. Another relationship between same entities descripted before is makes which I considered is occurs as member of staff payments to staff or performer. A carer can be a member of staff because in the text a particular member of staff is responsible to look after VIPs or actors. For the same reason I decided to implement relationship cares from Carer entity to the Actor entity. Furthermore, as described in the text in the project brief the cast has to belong to actor. For this reason, I included this specialisation relationship between them. The passage mentions that there is cast of a play. Consequently, I concluded that the relationship performs from entity cast to theatre plays. Moreover, as explained in the project brief the theatre play is a type of the performance. Given this, I included this specialisation relationship between them. A venue offers performance to customers. This relationship appears as text says” a number of venues all over the UK which offer performances to the public”. Additionally, as the brief mentions the Performances and Meal are types of events. Given this, I included two specialisation relationship between them. A band can play a concert. Evidence for this relationship is provided in the phrase” The details of the bands or musical ensembles which play concerts”. Customers book tickets was mentioned three times in the second paragraph. In addition, as the brief mentions the VIP is a kind of Customer. As a result, I included this specialisation relationship between them. Staff or performer payment have relationship receives to actor and band as “as fee payment is made to the whole band or group for performing”.

**4.** Provide an executive summary of how you identified the **multiplicity constraints** for PerfoSpace. In other words, explain and justify how you identified these multiplicities in a written narrative in the first person. This needs to fit on no more than one page of the report.

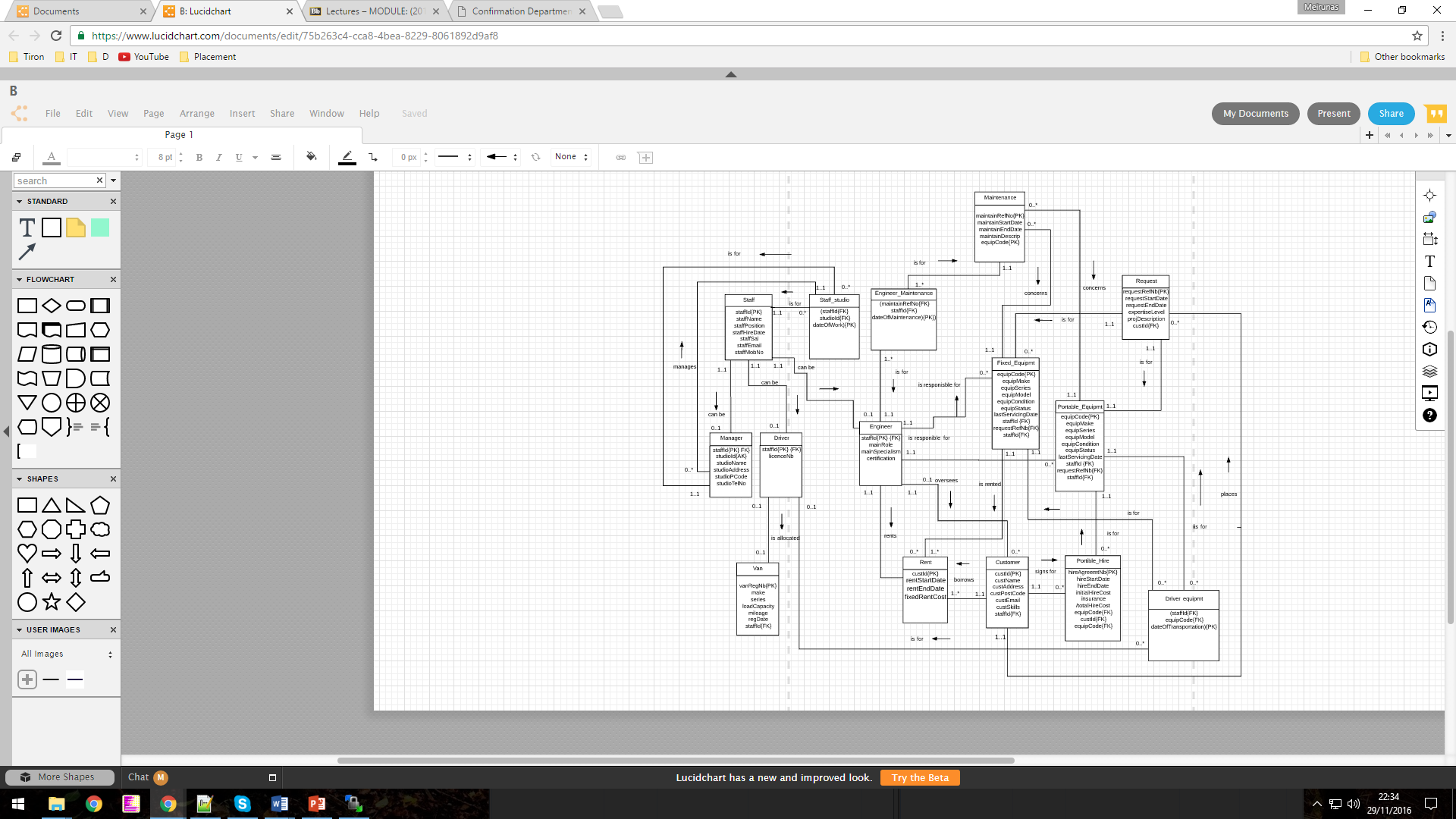
Multiplicity is Staff member one to one and one to many to Venue the reason for this company existing is one and can run one or more venues as mention in the following excerpt “company owns and operates a number of venues all over UK”. Company one to one as belong to one company which makes Staff or performer payment ten to many “it is essential for the payments of all fees to performers” and consider staff payment to necessary to be made. Staff member one to one as is for one staff which receives payment to Staff or performer payment which is get just one at the time. Actor which can be from one to many as mention in text” Upon request, some actors, perhaps lead roles or high profile performers, can be assigned to particular” and Carer one to one which assigned to actor. Carer is one to one as one carer can care for VIP evidence” allocated a personal member of staff”. Staff member zero to many as not each staff to Venue as staffs request to “work at the various venues run by the firm”. Cast one to one because one cast can perform Theatre play with range of zero to many” PerfoSpace offers performance such as Theatre plays”. Venue one to one because at one venue there are zero to many events in event Entity as in brief” Once they have selected the particular set of events”. Customer zero to many as there can be no customers at events which participating and many as company are creating register for them” To attend an event, customer need to register” and in Event one to one as existing one event for customer at a time. Concert zero to many as this company “operates a number of venues all over the UK” I decide they have got many concerts to have and Band zero to many as they got many different bands to play. Customer zero to many as they got large number of staff so that means they got and many customers which books zero to many tickets as they are available to customers. Customer makes one to one because “They have the option to book tickets only” and Ticket or membership multiplicity is zero to many because it can receive from zero to many payments. Actor one to one as “as separate fee payments are made to each individual member of the cast” and Staff or performer payment as it one between members. Staff or performer payment one to one as it belongs to group of staff and band receives one to one as it sends to one band. Event one to one multiplicity as one event is and has Ticket fifty to many as I after consideration decide to set to 50 as it minimum amount have to be printed of ticket but which can be expanded if request.

**5.** Provide an executive summary of how you identified the **attributes and primary keys** for PerfoSpace. In other words, explain and justify how you identified these attributes and primary keys in a written narrative in the first person. This needs to fit on no more than one page of the report.

I did choose minimum characters for primaries keys and a required attribute. I added ID as primary key for staff member to unique identify staff each employee should contain separated information about them as “PerfoSpace employs a large number of staff”. Role as each member have roles and name to contain names of staff. VIP attributes are ID primary key as it identifies uniquely each VIP which inherited from customer. I added fixed fee because in the saying “flat fee which is fixed by PerfoSpace”. Date and status attributes from text “The date and the status of the membership… need to be recorded”. Frequency as” can vary depending on year”. Cast primary key is name as each cast have unique cast name. Venue ID is primary key because each venue got unique address by which can be identified. Location provides place of venue and be meaningful to have as company have “all over UK”. Customer ID because each customer has his/her unique ID when is registered. These typical attributes which I identified from online and common sense which contains details of customers so I included too: name, address, contact number, email, DOB and Gender. VIP it has discount attribute because “scheme allows VIPs to get a %20 discount”. Performance type because have two typed “performance(concert) or (play)”. Age rating reason for this attribute is” if the age rating for the performance is suitable”. Title because each performance type has got title and genre which can be different. Relationship between Customer and Ticket have ID which is primary key as each booking have unique reference. Number of evenings “to attend one or more evenings”. Next attribute Number of evenings as from text “This needs to take into accounts the number of event”. In additional Accepted “if the age rating for the performance is suitable” and the booking be accepted if there are sufficient ticket available Payment entity ID is primary key as each payment got unique transfer number. Price because transfer cannot exist without price and available from” sufficient tickets available”. Ticket entity ID is primary key as each ticket purchased got ID. Price is necessary as ticket cannot exist without price. Available shows are available or not from “The booking be accepted if there are sufficient ticket available”. Meal type “on offer” and Menu “operate a standard menu. Band entity have name so this is reason for to have attribute. Style is because bands have got different styles and type of band as existing different types. Ticket or membership payment Discount attribute because ticket and membership can have discount “%20 discount”. Type of customer payment as customer can pay with different ways as in real life scenarios such as bank transfer or PayPal. Staff or performer payment attribute account number and sort code which have to exist in case the staff or performer receive payment. Event entity have got unique ID for each event. Date when is happens. Start time and End time to have necessary information about time. Description to find out easily what event offering.

**Part B**

**6.** Produce a complete **Logical Entity-Relationship Diagram** for SoundSpace. This needs to include all the **correct relations, relationships, multiplicity constraints, attributes, primary keys** and **foreign keys**. This needs to fit on one page of the report.



**7.** Provide an executive summary of how you produced the SoundSpace **logical ERD** i.e. how you **mapped** the SoundSpace conceptual ERD into a full **logical relational schema**. In other words, you need to refer to the Logical Data Modelling Methodology and explain in a written narrative in the first person what you need to do to resolve each relationship and derive the relevant relations (i.e. tables) with all their attributes, primary keys and foreign keys. This needs to fit on no more than two pages of the report

Staff and Studio entities relationship was many to many that means I have to apply rule 5 of design methodology. I created new table Staff\_studio and add primaries keys staffid and studioid of two tables to foreign keys and introduced new primary key dateOfWork to represent fact that this relationship takes place in several time points. I applied rule 10 Optional Or between Parent class to three children Manager, Driver and Engineer. The children inherited the primary key staffid and this became foreign key. Relationship undertakes stays and was before and many to many between Maintenance and Engineer so this mean I suppose to apply rule 5 again. This lead me to create new table Engineer\_maintenance with the primary keys of the previous relationships as foreign keys maintainRefNo staffId and together with the additional attribute dateOfMaintenance they constituted the primary key of the new table. For the relationship from the entity Maintenance to the entity Equipment I have chosen rule 1 as is represents one to many multiplicities. After this was figured out I transferred the primary key equipCode as foreign key to Maintenance as the rule required. Rent appears as a complex ternary relationship between entities Engineer, Customer and Fixed equipment, which means rule 6 should be applied. As the result I created a new entity called Rent connected with other three entities. The primary keys of the latter became the foreign keys of the former. Driver and Van entities have got 1 to 1. Based on multiplicities optional on both side as it not necessary to have driver with van**.** I included foreign key staffed in Van entity. Another step was Customer to Portable\_hire and it referenced with custid.

Equipment with two sub-entities was resolved into two tables following the rule regarding Generalisation with {Mandatory,Or}. The primary key of the original super entity became primary key of the sub-entities.

Driver transports relationship to equipment many to many. This is the reason I have chosen to apply rule 5. I kept the two original parent tables and added 1 link table (driver equipment) associated with them. The link child table have referenced staffId and equipCode as foreign keys. The primary key of the link table is the combination of the 2 Primary keys of the Parent Tables. Customer table got one to many so this means rule 1 have to be applied. I followed this rule and write primary key custid must be foreign key in request table.

In the following passage I am writing the relationship names in bold. One to one **manages** the original relationship from Manager to Studio now connects Manager to Staff\_studio. One to one optional on both sides **is allocated** relationship still connects Driver to Van. One to many **is responsible** **for** relationship now connects Engineer separately to Fixed\_Equipmt and Portable\_Equipmt. One to many **oversees** relationship still connects Engineer to Customer. Many to one **concerns** relationship now connects Maintenance seperatly to Fixed\_Equipmt and Portable\_Equipmt.Many to one **is for** relationship now connects Request separately to Fixed\_Equipmt and Portable\_Equipmt . One to many **signs for** relationship still connects Customer to Portable\_Hire. One to many **places** relationship still connects Customer to Request. **Can be** relationship from Staff to its children Manager, Driver and Engineer is derived from generalisation with optional or rule. **Rents**, **is rented** and **borrows** relationships were derived from the complex ternary relationship rents. The first connects Engineer to Rent. The second connects Fixed\_Equipmt to Rent. The third connects Customer to Rent. The two **is for** relationships from Staff\_Studio to Manager and from Staff\_studio to Staff were derived from the original many to many relationship works in from Staff to Studio.The two **is for** relationships from Engineer\_Maintenance to Maintenance and from Engineer\_Maintenance to Engineer were derived from the original many to many relationship works in from Engineer to Maintenance. The three **is for** relationships from Driver equipmt to Driver, from Driver equipmt to Fixed\_Equipmt and from Driver equipmt to Portable\_Equipmt where derived from original many to many relationship transports between Driver and Equipment.

1. Write some SQL code (DDL) to create the Studio table in the MySQL RDBMS. The studio id should store numbers. The studio name should be unique. No field should be left empty. The table should have a primary key. Both the code and a screenshot of the structure of the table should be included. The screenshot should show your student id number (right above the structure of the table). This needs to fit on one page of the report.

I created Studio table code as shown in screenshot below. The studio id storing number with INT data type. The studio name is primary key. Fields are not null because require do not have empty records.

CREATE TBALE Studio

(

studioId INT (6) not null,

studioName VARCHAR(50) not null,

studioAddress VARCHAR(50) not null,

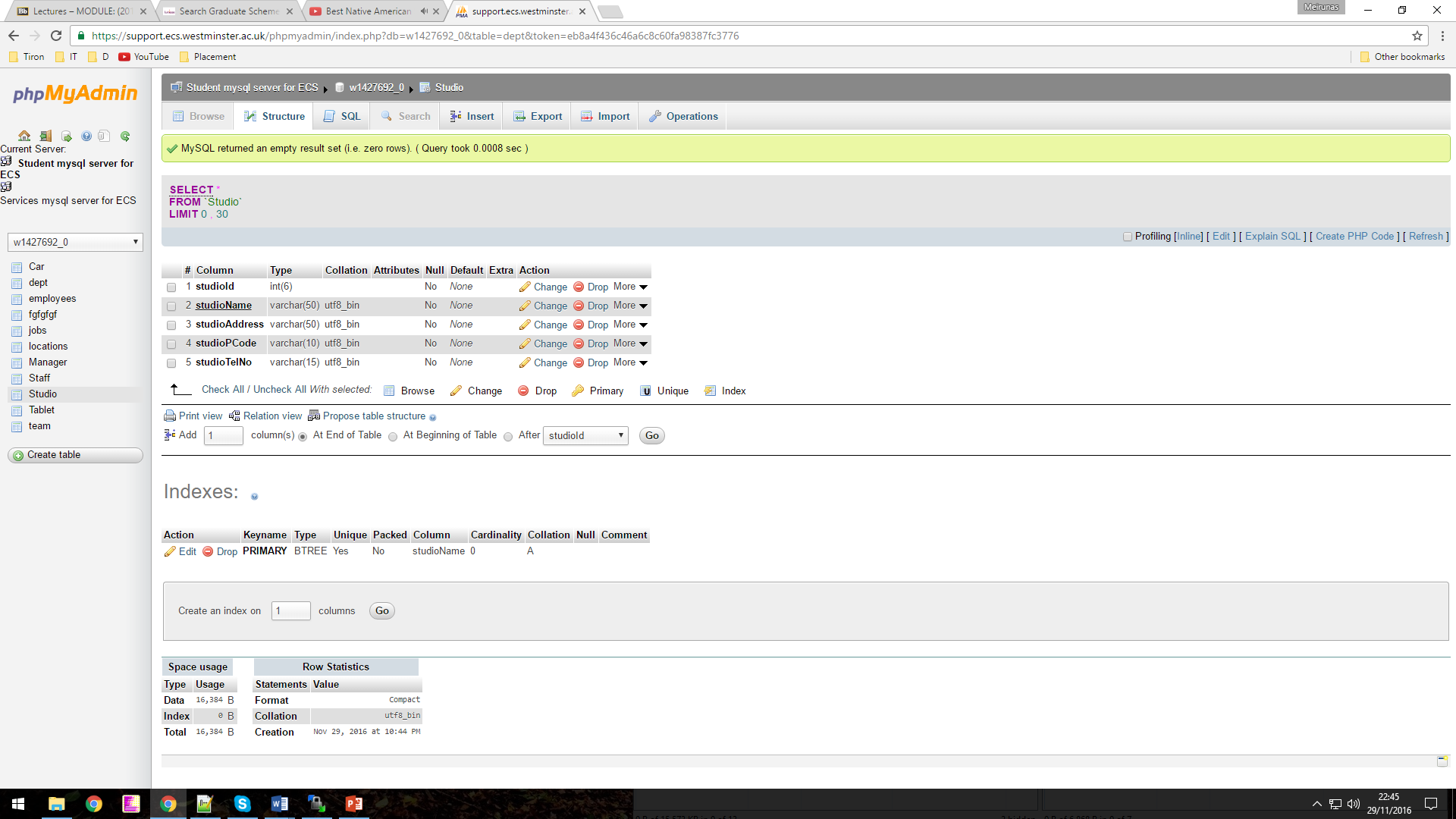
studioPCode VARCHAR(10) not null,

studioTelNo VARCHAR(15) not null,

constraint s\_sn\_pk PRIMARY KEY (studioName)

)

ENGINE=InnoDB;

Print screen shows structure of table.

1. Create a simple PHP file called **addstudio.php** to display a Web-based form to allow an administrator to enter the details of new studios onto the Studio table in the MySQL RDBMS. Your form should look like the one shown on figure 1. Both the **code** and a **screenshot of the Web page** should be included. The screenshot should show your student id number (in the URL bar). This needs to fit on one page of the report.

<?php

include ("db.php");

$pagename="Add a Studio";

echo "<title>".$pagename."</title>";

echo "<h2>".$pagename."</h2>";

echo "<body>";

//create a html form to capture user input

echo "<form method=post action=getstudio.php>" ;

echo "<table border=0 cellpadding=10>";

echo "<tr><td>\*Studio Id Number </td>";

echo "<td><input type=text name=form\_sIdnumber size=35></td></tr>";

echo "<tr><td>\*Studio Name </td>";

echo "<td><input type=text name=form\_sname size=35></td></tr>";

echo "<tr><td>\*Studio Address</td>";

echo "<td><input type=text name=form\_saddress size=35></td></tr>";

echo "<tr><td>\*Studio Postcode</td>";

echo "<td><input type=text name=form\_spcode size=35></td></tr>";

echo "<tr><td>\*Telephone Number</td>";

echo "<td><input type=text name=form\_stnumber size=35></td></tr>";

echo "<tr><td><input type=submit value='Add Studio'></td>";

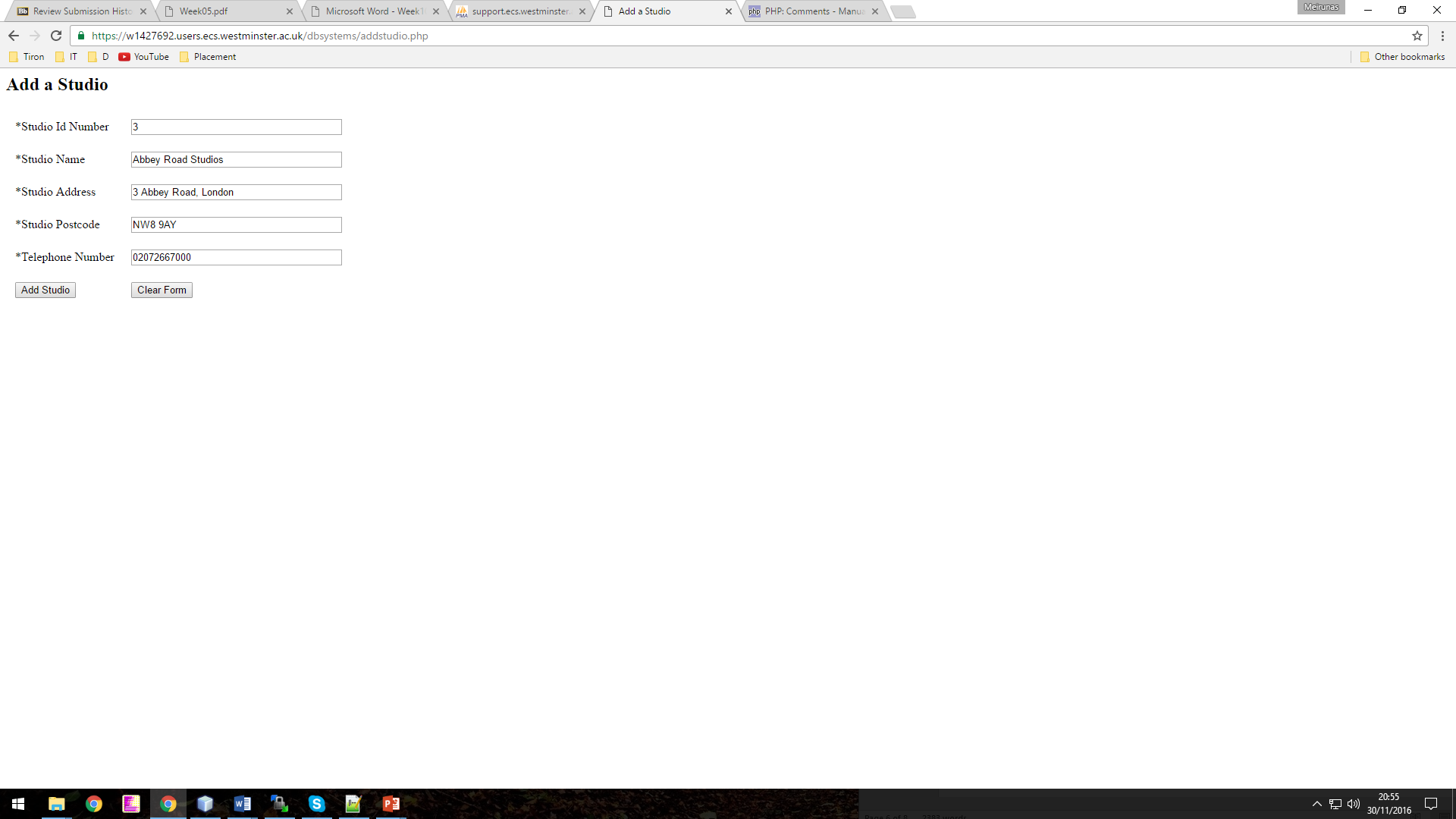
echo "<td><input type=reset value='Clear Form'></td></tr>";

echo "</table>";

echo "</form>" ;

echo "</body>";

?>



1. Create a simple PHP file called **getstudio.php** to display a Web-based confirmation page displaying the added studios. Your page should not only show the last added studio but also two studios added previously. Your confirmation page should look like the one shown on figure 2. Both the **code** and a **screenshot of the Web page** should be included. The screenshot should show your student id number (in the URL bar). This needs on no more than two pages of the report.

<?php

include ("db.php");

$pagename="Studios Confirmation";

echo "<title>".$pagename."</title>";

echo "<h2>".$pagename."</h2>";

echo "<body>";

//capture the value inserted in the form's fields and posted through here

//Store them in local variables

$stdidnum=$\_POST['form\_sIdnumber'];

$stdname=$\_POST['form\_sname'];

$stdaddress=$\_POST['form\_saddress'];

$stdpcode=$\_POST['form\_spcode'];

$stdtelnum=$\_POST['form\_stnumber'];

//check if any of the mandatory fields were not filled in

if (empty($stdidnum) or empty($stdname) or empty($stdaddress) or empty($stdpcode) or empty($stdtelnum))

{

echo "<p>Please ensure all fields with a \* are filled in!";

}

else

{

$addstdSQL=

"insert into

Studio (studioId, studioName, studioAddress, studioPCode, studioTelNo)

values (".$stdidnum.", '".$stdname."','".$stdaddress."','".$stdpcode."', '".$stdtelnum."')";

//run SQL query

$exeaddstdSQL=mysql\_query($addstdSQL);

//check error code and error message, can be commented out later

//echo "<p>Error code: ".mysql\_errno($conn);

//echo "<br>Error msg: ".mysql\_error($conn);

//else check individual error codes and display appropriate message

if (mysql\_errno($conn)!=0)

{

echo "<p>There is an error with the location you entered.";

//error code for breach of PK or unique constraint

if (mysql\_errno($conn)==1062)

{

echo "<br>The value entered for the new location id is not valid as it is already used.";

}

//error code for inserting character that is problematic for SQL query

if (mysql\_errno($conn)==1064)

{

echo "<br>Values entered for the location details are not valid.";

}

}

//write SQL query

$viewstdSQL="select \*

from Studio

order by studioId desc

limit 3";

//Run SQL query or exit if any errors are retrieved

$exeviewstdsSQL=mysql\_query($viewstdSQL) or die (mysql\_error());

echo "<hr>";

while ($arraystds=mysql\_fetch\_array($exeviewstdsSQL))

{

echo "<p> ".$arraystds['studioId'];

echo "<br>".$arraystds['studioName'].", ".$arraystds['studioAddress']." ".$arraystds['studioPCode'];

echo "<br>".$arraystds['studioTelNo'];

echo "<hr>";

}

}

echo "</body>";

?>

