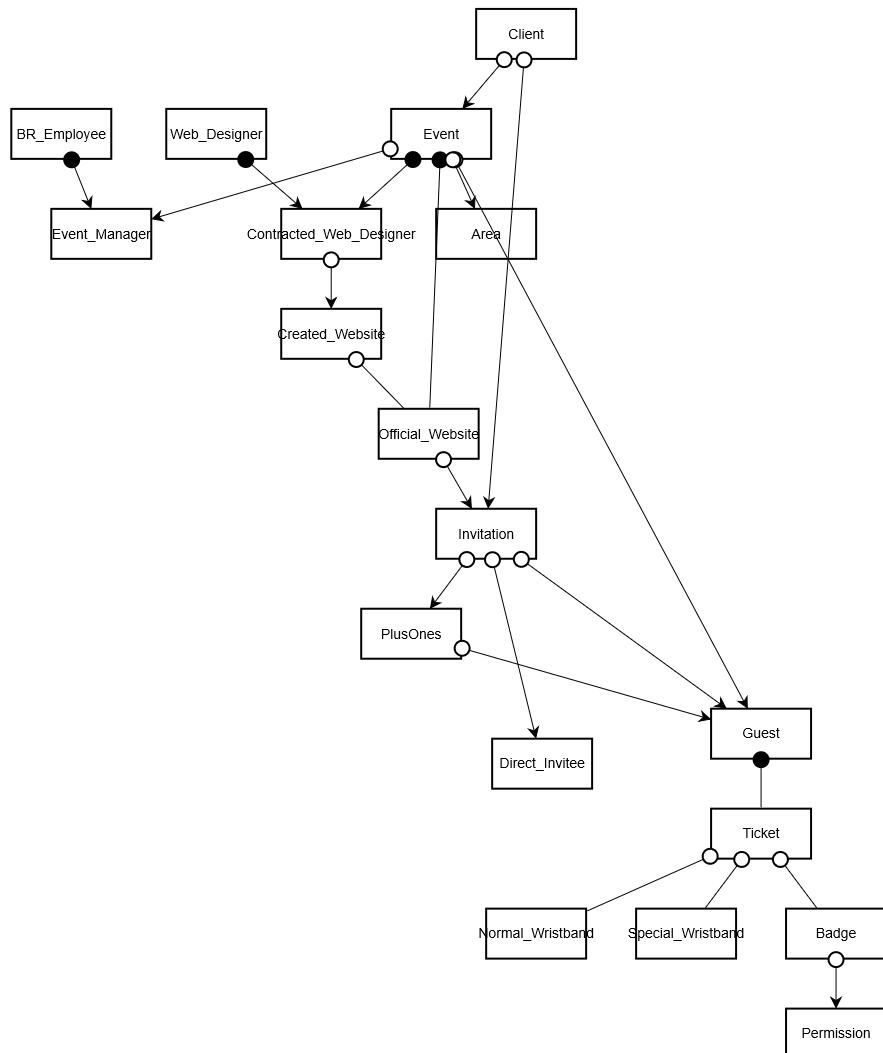
AMMIS GROUP REPORT - BLACK RABBIT TICKET SYSTEM

We started working on the assignment from the 21st of March. Knowing that the deadline for the submission of the peer review report was the 21st of April, we decided to follow the agile methodology and did our work in cycles of one week. At the end of each cycle we held a group meeting so that we could review the work which was done and see what changes had to be made and what we had done well in the previous cycle.

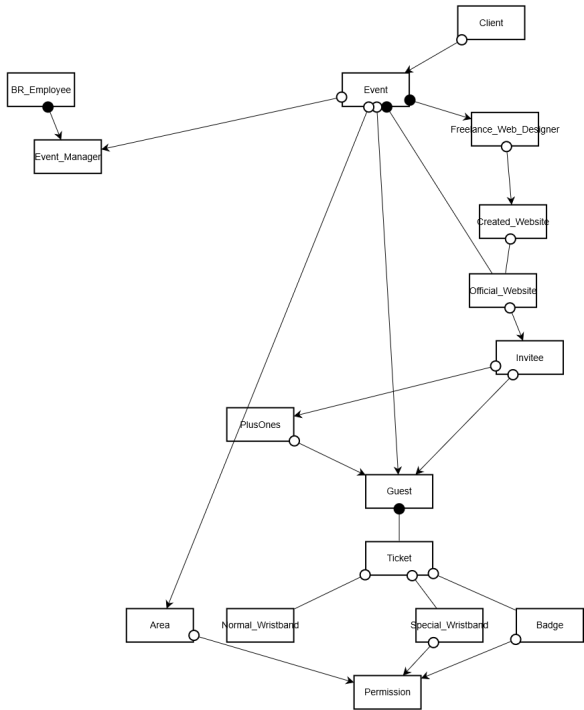
We also divided the work so that 2 members of our group designed the model, 2 members wrote the test cases, 1 member of our group tested the cases and 1 member wrote the report for every cycle.

**4th April**



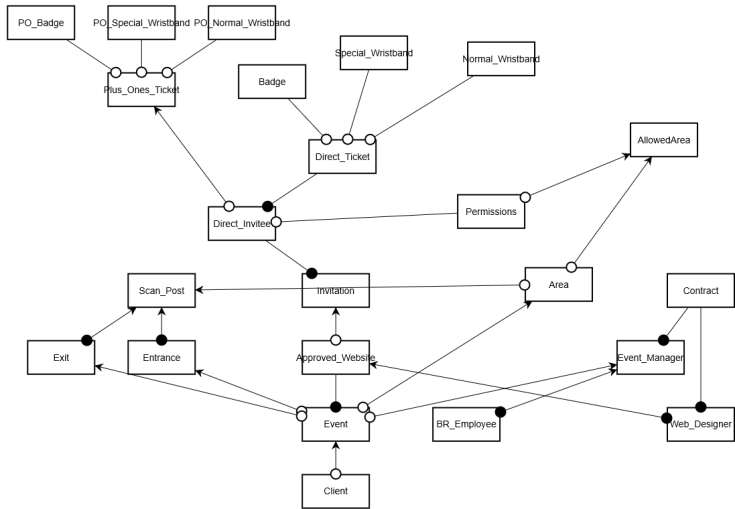
During the first cycle, we concentrated on getting familiar with the description of the working of the Black Rabbit. We made a list of the nouns in the text description which were potential business objects and a list of the verbs which could be construed as business events. The edg diagram above was the result. Our handling of the websites and the plus ones was not proper and we were not sure of the permissions part which we felt we would integrate better in the next cycles.

**11th April**



During the second cycle, we tried further refining the edg. We were still not able to correctly integrate the plus ones part into our model and were unsure how to approach it. Also the permissions part was still an issue and we made it exitence dependent on the area object too. In the previous cycle we had concentrated mainly on the edg but now we decided to pay some attention to the OET and the FSMs part.

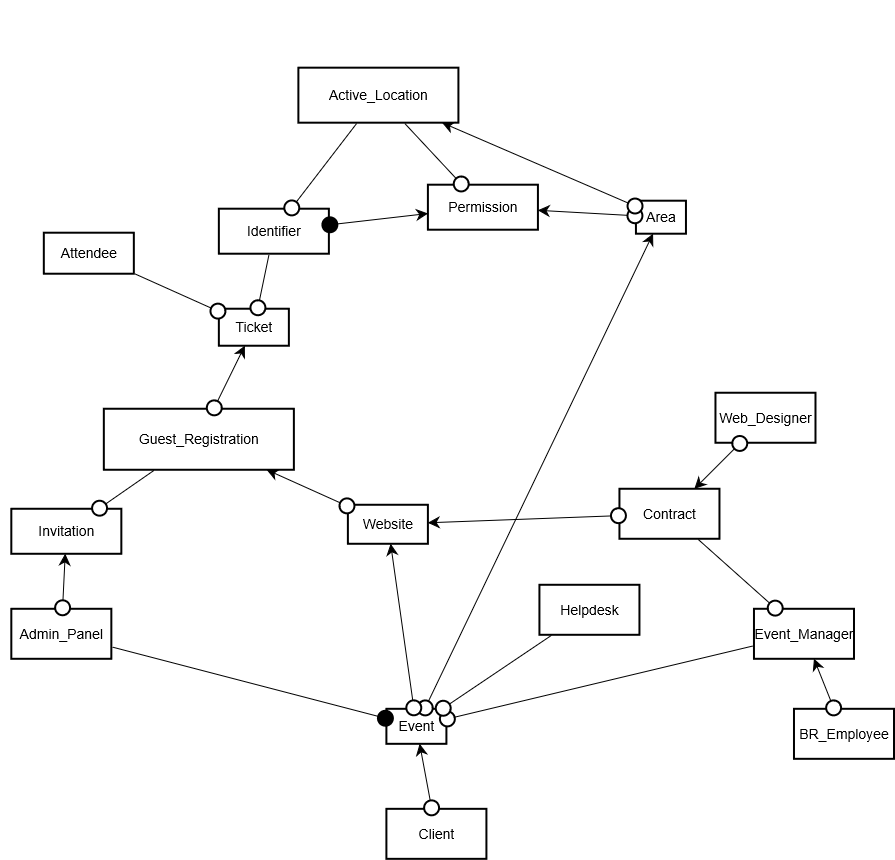
**21st April- Peer Review**



During the cycle just before the Peer Review we decided to add 2 new business object i.e. the exit and the entrance in order to make the permissions part of our model fit more cleanly with our model during the testing. Also we decided to split the Plus Ones ticket and the ticket of the person who i directly invited because there might be a difference in the prices and also in the permissions. We also introduced a new business object, AllowedArea to help with the permissions part. We also improved on the OETs and te FSMs and got a better understanding of how they all should fit together.

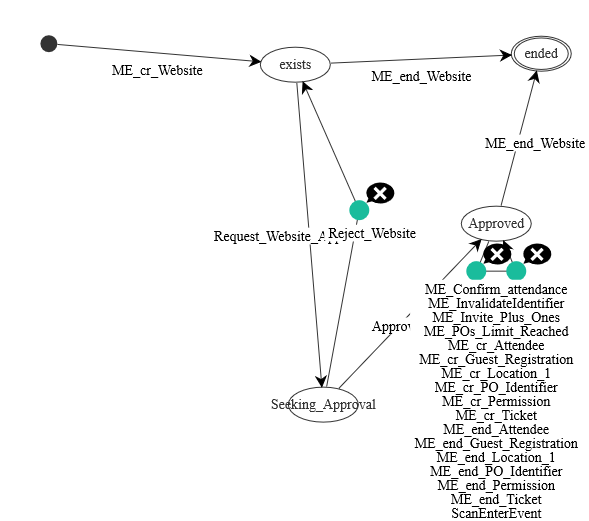
**Final Model – 15th May**

**EDG:**

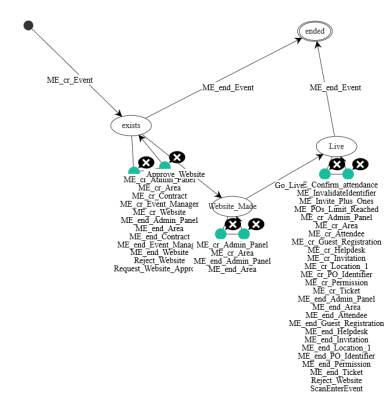


We decided to take some of the peer review suggestions into account. We also decided to revamp the registration process to integrate the direct tickets and the plus ones tickets which were earlier modelled separately. We also decided to revamp the way in which the permissions part was being modelled. We also changed a few of the cardinalities and also made a lot of chages in the fsms and the oet table.

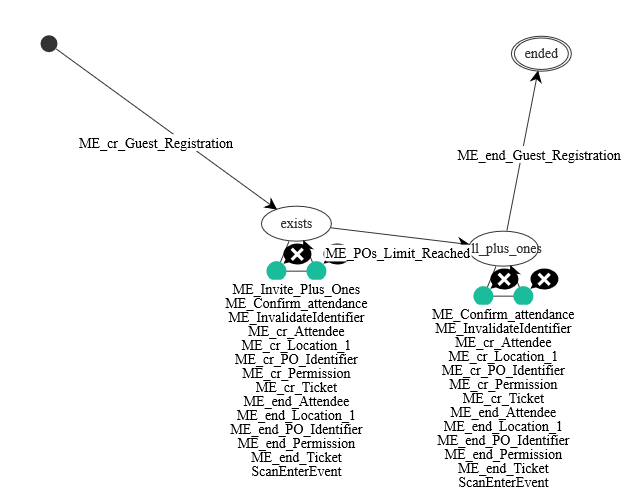
**Website FSM:**



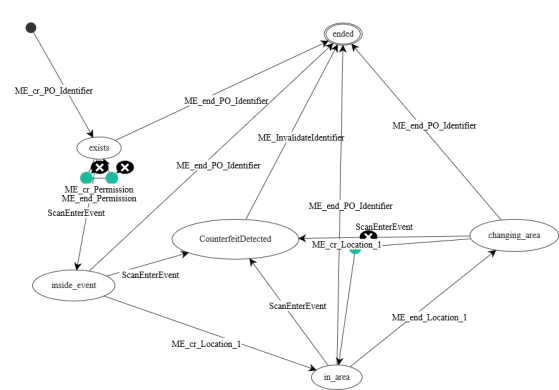
**Event FSM:**



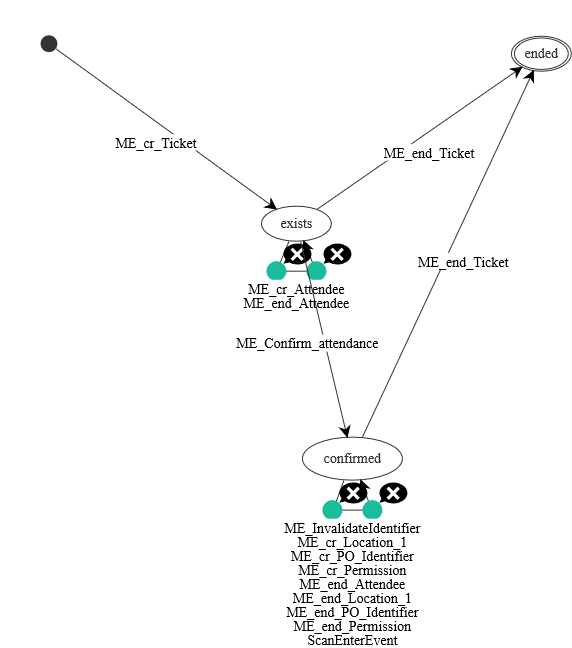
**Guest Registration FSM:**



**Identifier FSM:**



**Ticket FSM:**



**Business Objects:**

1. **Client :-** represents the clients of Black Rabbit.
2. **Event :-** represents the events that are conducted by Black Rabbit and is thus existence dependent on the Client object. *Attributes:-* Start\_Date, End\_Date, Name
3. **Admin\_Panel :-** is the method by which the clients can send invitations and differs from event to event according to our model.
4. **Invitation :-** represents the invitations sent out from the admin panel and is thus existence dependent on it. *Attributes:-* Price, No\_of\_PlusOnes, Plusone\_Fee
5. **Event Manager :-** represents the event manager for each event and is thus existence dependent on the BR\_Employee object and the event object.
6. **BR\_Employee :-** represents the employees of Black Rabbit.
7. **Helpdesk :-** represents the helpdesks set up around the event and is thus existence dependent on it.
8. **Contract :-** represents the choosing of the web designer and is done by the Event Manager of and event and is thus existence dependent on him/her.
9. **Web\_Designer :-** represents all the web designers from whom one is chosen in order to be the designated web designer for an event.
10. **Website :-** represents the website which is chosen for each event.
11. **Guest\_Registration :-** represents the registration process where based on an invitation, the invitee can go on the website and register himself/herself and also invite plus ones as required. It is thus existence dependent on the invitation and the website.
12. **Ticket :-** represents the tickets for both the directly invited people and the plus ones and is this existence dependent on the guest registration object. *Attributes:-* StartDate, EndDate, Price, Layout, OtherPermissions
13. **Attendee :-** represents the list of people who actually attend the event. It is thus existence dependent on the ticket with a optional one relationship as people who have the ticket can decide to not attend the event too. *Attributes:-* Name, email
14. **Identifier :-** represents the object that is got during the event in return for the ticket and which is required for going from one area to another. *Attributes:-* Name, UniqueCode
15. **Permissions :-** represents the list of areas which each identifier has access to. It is thus existence dependent on the area and the identifier.
16. **Area :-** represents the list of areas in an event and is thus existence dependent on the event.
17. **Active\_Location :-** gives a log of all the times that a particulat identifier was attempted to be used to move from one area to another. It is thus existence dependent on the area object, the permissions object and the identifier object.

**Business Events:**

|  |
| --- |
| 1. **EV\_cr\_Event :-** represents the creation of an event. |
| 1. **EV\_end\_Event :-** represents the end of an event. |
| 1. **EV\_cr\_Client :-** represents the creation of a client. |
| 1. **EV\_end\_Client :-** represents the end of a client. |
| 1. **EV\_cr\_BR\_Employee :-** represents the creation of a Black Rabbit Employee. |
| 1. **EV\_end\_BR\_Employee :-** represents the end of a Black Rabbit Employee. |
| 1. **EV\_cr\_Event\_Manager :-** represents the creation of a Event Manager. |
| 1. **EV\_end\_Event\_Manager :-** represents the end of a Event Manager. |
| 1. **EV\_cr\_Web\_Designer :-** represents the creation of a Web Designer. |
| 1. **EV\_end\_Web\_Designer :-** represents the end of a Web Designer. |
| 1. **EV\_cr\_Area :-** represents the creation of an Area. |
| 1. **EV\_end\_Area :-** represents the end of an Area. |
| 1. **EV\_cr\_Invitation :-** represents the creation of an Invitation. |
| 1. **EV\_end\_Invitation :-** represents the end of an Invitation. |
| 1. **EV\_cr\_Website :-** represents the creation of a Website. |
| 1. **EV\_end\_Website :-** represents the end of a Website. |
| 1. **EV\_cr\_Contract :-** represents the creation of a Contract. |
| 1. **EV\_end\_Contract :-** represents the end of a Contract. |
| 1. **EV\_cr\_Ticket :-** represents the creation of a Ticket. |
| 1. **EV\_end\_Ticket :-** represents the end of a Ticket. |
| 1. **EV\_cr\_PO\_Identifier :-** represents the creation of an Identifier. |
| 1. **EV\_end\_PO\_Identifier :-** represents the end of an Identifier. |
| 1. **Go\_Live :-** represents the event going live. |
| 1. **EV\_cr\_Helpdesk :-** represents the creation of a Helpdesk. |
| 1. **EV\_end\_Helpdesk :-** represents the end of a Helpdesk. |
| 1. **EV\_cr\_Admin\_Panel :-** represents the creation of a Admin Panel. |
| 1. **EV\_end\_Admin\_Panel :-** represents the end of a Admin Panel. |
| 1. **EV\_cr\_Guest\_Registration :-** represents the creation of a Guest Registration object. |
| 1. **EV\_end\_Guest\_Registration :-** represents the end of a Guest Registration object. |
| 1. **EV\_POs\_Limit\_Reached :-** represents that the max number of Plus ones that can be invited in each Guest Registration process has been reached. |
| 1. **EV\_Invite\_Plus\_Ones :-** represents the plus ones that are being invited during the Guest Registration Process. |
|  |
|  |
| 1. **EV\_cr\_Attendee :-** represents the creation of an Attendee object. |
| 1. **EV\_end\_Attendee :-** represents the end of an Attendee object. |
| 1. **EV\_cr\_Permission :-** represents the creation of a Permissions object. |
| 1. **EV\_end\_Permission :-** represents the end of an Permissions object. |
| 1. **Reject\_Website :-** represents the event where a proposed website is rejected |
| 1. **Request\_Website\_Approval :-** represents the event where a created website is sent for approval. |
| 1. **Approve\_Website :-** represents the event where a proposed website is approved. |
| 1. **ScanEnterEvent :-** represents the time when the identifier is scanned at a scan post. |
| 1. **InvalidateIdentifier :-** represents the time when a potential counterfeit ticket is confirmed to be a counterfeit ticket. |
| 1. **EV\_cr\_Location\_1 :-** represents the creation of an Active\_Location object. |
| 1. **EV\_end\_Location\_1 :-** represents the end of an Active\_Location object. |
| 1. **Confirm\_attendance :-** represents when a potential attendee confirms he/she is attending the evnt. |

# Answers to the Intermediate Peer Review Report:-

**EDG Review:**

**Existence Dependency Reasoning:**

1. Scan\_Post is dependent on Exit, Entrance and Area. Will every event require all three of these?

**Answer:-** None of the business objects exist anymore.

1. Aproved\_Website is the master of the Invitation- What happens if the client does not require a Website, how will the process of the Invitations happen?

**Answer:-** The Invitation object is now existence dependent on the Admin\_Panel object, not the website.

1. Direct\_Ticket/Plus\_Ones\_Tickets are the master of Normal\_Wristband, Special\_Wristband and Badge. We think the three dependent Objects are redundant and create unnecessary duplicates and can be merged into a single Object type.

**Answer:-** Yes. We took this advice and implemented it.

**Minimum 1 cardinality problems:**

1. Exit + Entrance - Scan\_Post: Scan post is a mandatory slave of both entrance and exit, while it is possible that scan post only exists in an entrance but not at the exit (because there is the possibility to not have an exit at all). How will you model this, so this functionality is possible?

**Answer:-** None of the 3 objects are present anymore.

1. BR\_Employee - Event\_Manager: This means that exactly when the employee exists it needs to be an EventManager. And this implies that the employee needs to be constantly being a manager throughout all their existence. Does this always hold true in reality?

**Answer:-** We took your advice and made the relationship optional.

1. Event\_Manager - Contract: A contract is not mandatory; it is an optional one. Thus, it should not be mandatory to have a contact, it should be an optional one.

**Answer:-** We took your advice and made the relationship optional.

1. Event - Approved\_Website: A client has the option to ask for a website or not, thus, an event might not need to have a website. A mandatory one cardinality blocks this.

**Answer:-** It is mentioned in the text that the website will be used in the invitation and registration process.

1. Invitation - Direct\_Invitee - Direct\_Ticket: An invitation does not always produce a participant.Here, the invitation must have a direct invitee who must also have a direct ticket. This implies all invitees must have at least one ticket, while invitees should be able to decline to participate in the event (i.e. no tickets). A possible solution would be changing the mandatory 1 relationship or differentiate the concept between Invitees and Attendees.

**Answer:-** We have remodelled the entire process of registration so it is very different now.

**Missing objects:**

1. Invoice - a ticket might have purchasing tickets, thus, to be able to send a ticket you need to put a constraint that in order to issue the e-ticket, the invoice has to be paid first. This payment problem however might not necessitate an invoice object, it depends on how your group wants to model it.

**Answer:-** We modelled it in a different way which we believe does not require the invoice object.

1. Admin Panel - not all events have an website where the registration process happens, thus the admin panel should work as a functionality for the registration process

**Answer:-** We took your advice and included the Admin Panel object in our model.

1. Helpdesk - needs to be able to make modifications in case of counterfeits (can create new tickets and can modify existing ones).

**Answer:-** We took your advice and included the Helpdesk object in our model.

**FSM Review**

The default FSM for many objects may be sufficient for most of the Object Types, however, for some of them consider whether additional states are necessary to specify conditions under which events can occur, this also allows for better record-keeping. Some FSM we advise you to reconsider modifying revolve around:

1. What we explained in the section: Missing object→ Invoice.

**Answer:-** We changed the registration process into something that we believe does not require the invoice object.

1. Helpdesk Object type: Create how the helpdesk can manage to change/modify existing Tickets

**Answer:-** We took your advice and included the Helpdesk object in our model. However, the creation and modification of tickets when couterfeits are detected has been handled in the ticket fsm itself.

1. Adding guards regarding budgetary constraints when choosing a website.

**Answer:-** We felt that this advice is not required taking the text description into account.

**Testing Problems:**

1. Invitation has a NoOfPlusOnes attribute which is supposed to be a constraint for the Plus\_Ones\_Ticket, however, we were able to create more than the allowed NoOfPlusOnes per Direct\_Invitee.

**Answer:-** In our new model we have modelled the constraint part in the Guest Registration fsm which takes into account the evnt when the max number of plus ones is reached.

1. Direct\_Ticket object does not allow for multiple day events.

**Answer:-** We no longer have a Direct\_Ticket object. The ticket object on the other hand has both a start\_date attribute and an end\_date attribute.