



Conference Management System

First Delivery: Report

20th November 2020

Information Systems Engineering

MIB - Biomedical Engineering Branch

Guilherme Lucas Peralta

up201704277@fe.up.pt

Practical class B

João Carlos Ramos Gonçalves de Matos

up201704111@fe.up.pt

Practical class A

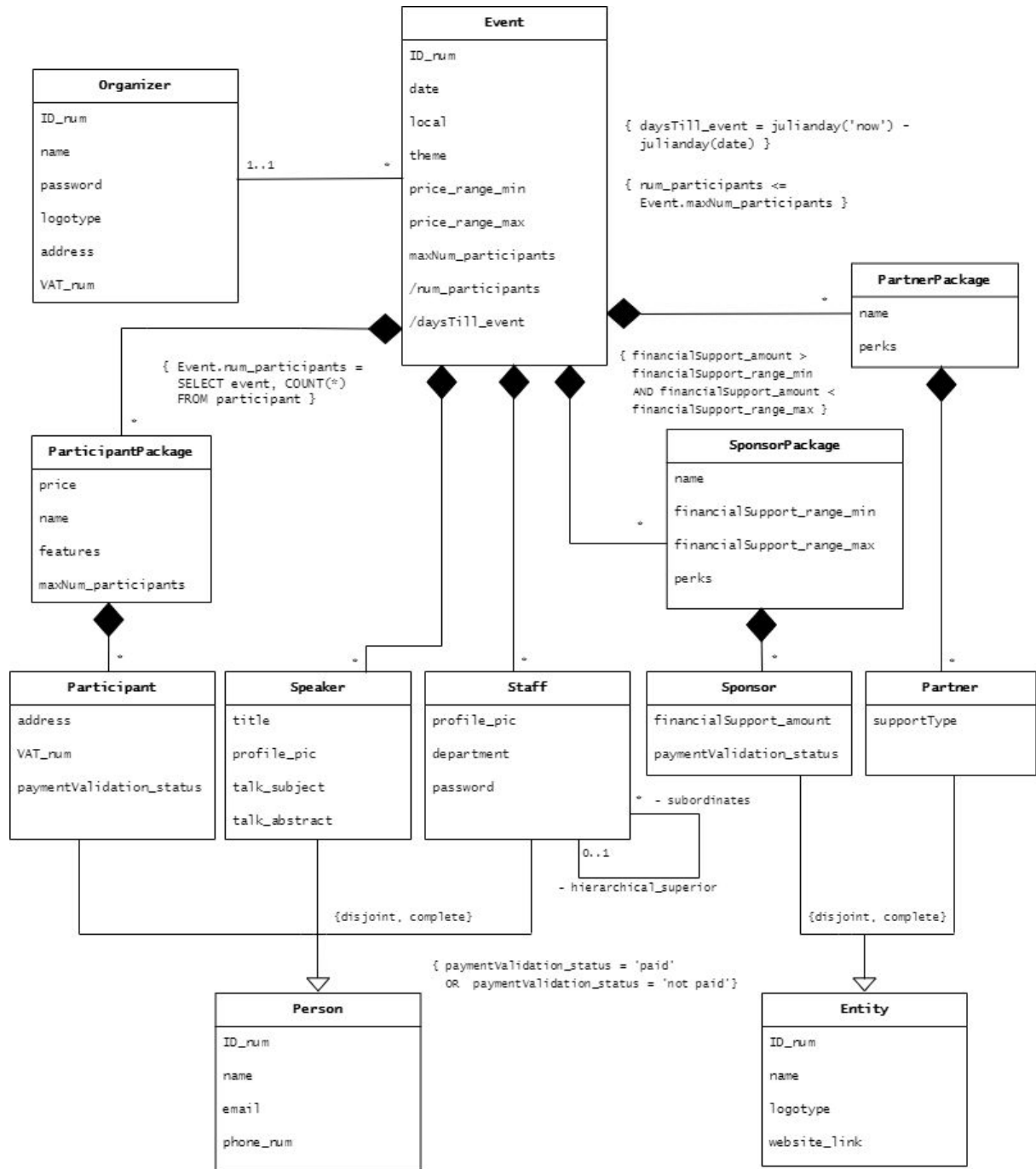
1. Description of the topic

A tech company wants to create a platform to automate the management of conferences. The platform is expected to announce events and support the logistics of conferences, by allowing the registration of participants, speakers, partners, sponsors and staff. Taking these features in consideration, the company elaborated the following list of requirements:

- The website must have a public interface, for announcement of events and registration, and a private one, exclusive for the event organizer and staff members, for management of the conference.
- The private interface must display relevant data, such as payments tracking, and lists of participants, speakers, staff, sponsors, partners.
- The **organizer** must have a name, password, address, and VAT number. A logotype is optional. A unique identifying number should be generated to each one of them.
- Each organizer can create as many events as wished.
- Each **event** must have an organizer, date, location, theme, price range, and (current) number of participants. A maximum number of participants can also exist. A unique identifying number should be given to each event.
- If existent, the maximum number of participants must be higher than the (current) number of participants.
- The event can have as many registration packages as wished, for participants, sponsors, and partners
- Each **participant package** must be associated with an existing event, and have a unique name, price, and features. A maximum number of participants per package is optional.
- Each **sponsor package** must be associated with an existing event, and have a unique name, range of financial support (minimum and maximum values), and associated perks.
- Each **partner package** must be associated with an existing event, and have a unique name and associated perks.

- When it comes to registration, there are 3 different roles for persons: participant, speaker, and staff. Each **person** must have a unique identifying number, name, email, and phone number.
- For the registration of supportive entities, there are 2 types of **entity**: sponsor and partner. Each entity must have a unique identifying number, name, logotype. A website link may be inserted.
- Additionally, each:
 - **participant** must be associated with an existing event and have an address, a VAT number, and a chosen package. If the event is paid, a payment validation status attribute is created, set not paid as default.
 - **speaker** must be associated with an existing event and have a title, profile picture, subject of talk, and abstract.
 - **staff** member must be associated with an existing event and have a department and a password. A profile picture, existence of subordinates, and hierarchical superior are optional.
 - **sponsor** must be associated with an existing event, have a chosen package, a financial amount of support, and a payment validation status (initialized as not paid). The financial amount of support must be according to the chosen package.
 - **partner** must be associated with an existing event, have a chosen package, and a support type.
- If an event is deleted, all the associated classes must be deleted too, except the organizer.
- The public announcement of the event should include a list of speakers, staff, validated sponsors and partners, as well as the date, location and available registration packages.
- As an extra, it would be nice to have a public interface with a search feature implemented, with the possibility of looking for all existing events, sorted by theme, organizer, speaker, dates, localization and price range.

2. UML Classes Diagram



3. Relational Model of the Database

Organizer(ID_num, name, password, logotype, address, VAT_num)

NOT NULL(name, password, address, VAT_number)

UNIQUE(VAT_number)

Event(ID_num, date, local, theme, price_range_min, price_range_max,
num_participants, maxNum_participants, daysTill_event, organizer -> Organizer)

NOT NULL(date, local, theme, price_range_min, organizer)

DEFAULT(num_participants = 0)

DEFAULT(price_range_min = 0)

CHECK(price_range_min >= 0)

CHECK(price_range_max = null OR price_range_max > price_range_min)

CHECK(maxNum_participants = null

OR num_participants < maxNum_participants)

ParticipantPackage(name, event -> Event, price, features, maxNum_participants)

NOT NULL(price, features)

PartnerPackage(name, event -> Event, perks)

NOT NULL(perks)

SponsorPackage(name, event -> Event, financialSupport_range_min,
financialSupport_range_max , perks)

NOT NULL(financialSupport_range_min, financialSupport_range_max, perks)

CHECK (financialSupport_range_min > financialSupport_range_max)

Person(ID_num, name, email, phone_num)

NOT NULL(name, email, phone_num)

UNIQUE(email, phone_num)

Participant(ID_num -> Person, address, VAT_num, paymentValidation_status,
package-> ParticipantPackage, event -> ParticipantPackage)

NOT NULL (address, VAT_num, package, event)

UNIQUE (VAT_number)

DEFAULT (paymentValidation_status = 'not paid')

Speaker(ID_num -> Person, event -> Event, title, profile_pic, talk_subject,
talk_abstract)

NOT NULL(event, title, profile_pic, talk_subject, talk abstract)

Staff(ID_num -> Person, event -> Event, profile_pic, department, password,
subordinates -> Staff, hierarchical_superior -> Staff)
NOT NULL (event, department, password)

Entity(ID_num, name, logotype, website_link)
NOT NULL (name, logotype)
UNIQUE (name, logotype, website_link)

Sponsor(ID_num -> Entity, financialSupport_amount, paymentValidation_status,
package -> SponsorPackage, event -> SponsorPackage)
NOT NULL (financialSupport_amount, paymentValidation_status, package, event)
CHECK(financialSupport_amount > 0)
DEFAULT (paymentValidation_status = 'not paid')
CHECK (Sponsor.financialSupport_amount >
 SponsorPackage.financialSupport_range_min
 AND Sponsor.financialSupport_amount <
 SponsorPackage.financialSupport_range_max)

Partner(ID_num -> Entity, supportType, package -> PartnerPackage, event ->
PartnerPackage)
NOT NULL (supportType, package, event)