

FEUPBook

Relatório BDAD 2019/2020

2MIEIC04

Eduardo Correia
up201806433

Ricardo Fontão
up201806317

João Diogo
up201806779

May 17, 2020

Contents

1	Introdução	3
2	Especificação	4
2.1	Publicador	4
2.1.1	Utilizador	4
2.1.2	Página	4
2.2	Conversa	5
2.3	Multimédia	5
2.4	Atividade	5
2.4.1	Publicação	5
2.4.2	Mensagem	5
2.4.3	Comentário	6
2.5	Reação	6
2.6	Evento	6
2.6.1	Grupo	6
3	Modelo concetual	7
4	Modelo relacional	8
4.1	Dependências funcionais	8
4.2	Análise Forma Normal	10
5	Restrições	11
5.1	Publisher	11
5.2	User	11
5.3	Page	11
5.4	ChatParticipant	11
5.5	Friendship	12
5.6	EventParticipant	12
5.7	PageFollower	12

5.8	GroupMember	12
5.9	Group	12
5.10	Event	12
5.11	Multimedia	13
5.12	Audio	13
5.13	Image	13
5.14	Video	13
5.15	Reaction	13
5.16	Comment	14
5.17	Post	14
5.18	Message	14
5.19	Activity	14
5.20	Chat	14

Chapter 1

Introdução

O nosso projeto consiste numa rede social denominada *FEUPBook* (inspirado pela já existente rede social, *Facebook*). Nesta, utilizadores poderão criar uma conta pessoal para falar uns com os outros, através da troca de mensagens de texto e ver diversas publicações do seu interesse no seu *feed* (a página inicial), bem como organizar **eventos**, juntarem-se em **grupos** ou edificar uma **página** dedicada a algum assunto em particular.

Chapter 2

Especificação

2.1 Publicador

Esta classe tem como função identificar quem são os elementos que podem realizar publicações, registar que publicações efetuaram e atribuir-lhes um nome.

2.1.1 Utilizador

Esta classe representa um **membro** da rede social. Este possui um id único para o identificar (correspondente ao último campo do *url* do seu perfil).

Um utilizador possui ainda vários outros dados pessoais (que o utilizador pode optar por não preencher por motivos de privacidade): número de telemóvel, data de nascimento, género (masculino ou feminino), morada (rua e localização).

Um **utilizador**, por norma, terá diversas **amizades** com outros utilizadores, para que possa aceder aos conteúdos do seu perfil, conversar mais facilmente com ele e ver as publicações no seu feed.

Para esse efeito, terá de enviar um pedido de amizade ao outro, o qual pode ser **aceite** ou não. Se for aceite, a ligação **pedido de amizade** deixa de existir e passa a ser uma ligação de **amizade**. Se for rejeitada, essa ligação deixa de existir sem ser substituída por nenhuma outra a um novo **pedido de amizade** surgir por parte de um dos dois utilizadores.

Tanto uma **amizade** como um **pedido de amizade** registam a data em que foram efetuados.

2.1.2 Página

Esta classe representa uma **página** na nossa rede social.

Esta tem também associada a si várias publicações.

Os utilizadores relacionam-se com um página na medida em que um deles é o seu administrador (normalmente correspondente ao criador da página, mas este cargo pode ser transferido) e possui vários seguidores, utilizadores que colocaram gosto na página e vêem as publicações no feed.

Tem um utilizador que é o seu administrador e pode ter muitos seguidores(utilizadores).

2.2 Conversa

Numa **rede social** é indispensável a capacidade de os utilizadores conversarem entre si, como tal estes podem agregar-se numa conversa (que possui no mínimo 2 utilizadores).

Uma **conversa** é composta por diversas mensagens.

Ainda neste contexto, cada utilizador pode possuir uma alcunha própria.

2.3 Multimédia

A rede social do nosso projeto possui a capacidade de partilhar ficheiros **multimédia**, quer seja através de mensagens ou publicações e dividem-se essencialmente em três **categorias**, áudio, imagem e vídeo. Cada ficheiro destes possui um título, correspondente ao seu nome em memória, bem como um url que indica a localização do ficheiro para lhe aceder.

É possível, no entanto, enviar qualquer tipo de ficheiros, não só áudio, imagem ou vídeo, porém se a sua extensão não corresponder a nenhum deste tipo de ficheiros, será enviado como um ficheiro binário que o utilizador pode descarregar.

Existe ainda um limite máximo do tamanho ficheiro que o utilizador pode enviar e é de 25 MB.

Tanto o **áudio** como **vídeo** possuem um comprimento da sua duração em segundos.

2.4 Atividade

Vários conteúdos na rede social terão **reações** por parte dos **utilizadores**, como tal, esta classe tem o objetivo de agregar esses mesmos conteúdos e manter o registo das reações que possuem.

Cada um desses conteúdos possui texto a acompanhá-los e a data em que foram efetuados.

2.4.1 Publicação

Não só com **mensagens** é possível comunicar na rede social, mas também com **publicações**. Possuem o benefício de não serem tão privadas (não precisam de ter um destinatário em particular, ficando publicadas no perfil do **publicador** em questão que seja o autor da **publicação**).

Esta pode ser composta por multimédia (áudio, foto ou vídeo) de tipos iguais ou diferentes.

Agrega ainda vários comentários feitos por utilizadores que pretendam demonstrar a sua opinião em relação à mesma.

2.4.2 Mensagem

Uma **mensagem** é o modo como os utilizadores comunicam numa conversa.

Pode ser uma mensagem de texto, a partilha de um ficheiro multimédia ou ambos, ou seja, um utilizador pode enviar um ficheiro multimédia com descrição (texto), sem descrição, ou só uma mensagem de texto.

2.4.3 Comentário

Um **comentário** é feito por um **utilizador** a uma **publicação** como modo de iniciar uma discussão sobre a publicação ou simplesmente realizar algum tipo de observação/denotação.

2.5 Reação

Um utilizador nem sempre tem de se manifestar com mensagens ou comentários, podendo optar por simplesmente deixar uma **Reação**. Para tal, terá a opção de escolher uma das seguintes **reações**.



Like



Love



Haha



Wow



Sad



Angry

2.6 Evento

Se um **utilizador** desejar, pode criar um **evento** para marcar um **acontecimento** relevante e convidar outros utilizadores a participarem no mesmo.

Um **evento** possui um nome que o identifica, bem como uma breve descrição do que se trata e, possivelmente, o local onde se realiza, e a data da sua realização.

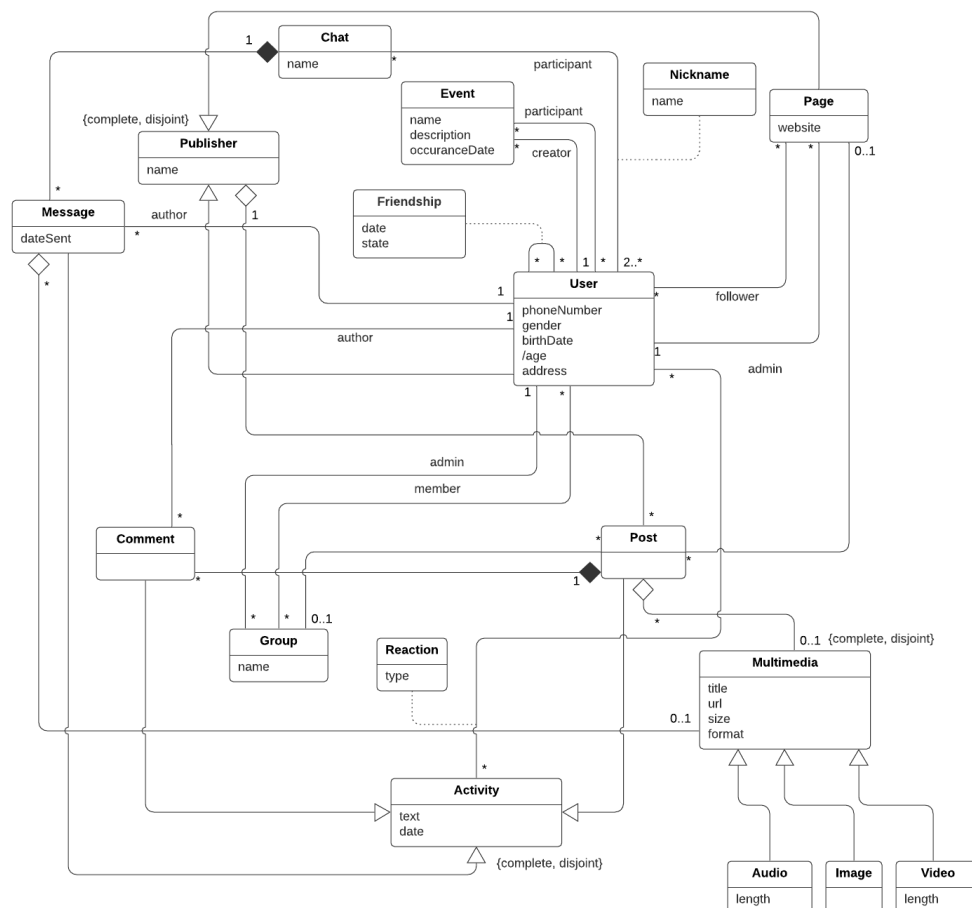
Por definição, a reação padrão será o *like*.

Um **utilizador** apenas pode reagir com uma das possíveis reações a um post ou comentários, mas estes podem ter reações de vários utilizadores, inclusive dos seus autores.

2.6.1 Grupo

Um grupo possui como função agregar diversos utilizadores com um interesse em comum. Um utilizador fazendo parte de um grupo, está habilitado a fazer uma publicação, estando esta visível para os restantes membros do grupo. Um grupo tem um utilizador como administrador.

Modelo concetual



Chapter 4

Modelo relacional

4.1 Dependências funcionais

Publisher(publisherID, name)

$\text{publisherID} \rightarrow \text{name}$

User(userID \rightarrow Publisher, phoneNumber, gender, birthDate, age, address)

$\text{userID} \rightarrow \text{phoneNumber}, \text{gender}, \text{birthDate}, \text{age}, \text{address}$

$\text{birthDate} \rightarrow \text{age}$

Friendship(senderID \rightarrow User, receiverID \rightarrow User, state, date)

$\text{senderID}, \text{receiverID} \rightarrow \text{state}, \text{date}$

Page(pageID \rightarrow Publisher, website, adminID \rightarrow User)

$\text{pageID} \rightarrow \text{website}, \text{adminID}$

PageFollower(followerID \rightarrow User, pageID)

Group(groupID, name, adminID \rightarrow User)

$\text{groupID} \rightarrow \text{name}, \text{adminID}$

GroupMember(memberID \rightarrow User, groupID \rightarrow Group)

Chat(chatID, name)

$\text{chatID} \rightarrow \text{name}$

ChatParticipant(participantID \rightarrow User, chatID \rightarrow Chat, nickname)

$\text{participantID}, \text{chatID} \rightarrow \text{nickname}$

Multimedia(multimediaID, title, uri, size, format)

$\text{multimediaID} \rightarrow \text{title}, \text{uri}, \text{size}, \text{format}$

$\text{uri} \rightarrow \text{title}, \text{size}, \text{format}$

Audio(audioID \rightarrow Multimedia, length)

audioID \rightarrow length

Image(imageID \rightarrow Multimedia)

Video(videoID \rightarrow Multimedia, length)

videoID \rightarrow length

Activity(activityID, text, date)

activityID \rightarrow text, date

Message(messageID \rightarrow Activity, dateSent, multimediaID \rightarrow Multimedia, authorID \rightarrow User, chatID \rightarrow Chat)

messageID \rightarrow dateSent, multimediaID, authorID, chatID

Post(postID \rightarrow Activity, publisherID \rightarrow Publisher, multimediaID \rightarrow Multimedia, pageID \rightarrow Page, groupID \rightarrow Group)

postID \rightarrow publisherID, multimediaID, pageID, groupID

Comment(commentID \rightarrow Activity, authorID \rightarrow User, postID \rightarrow Post)

commentID \rightarrow authorID, postID

Reaction(activityID \rightarrow Activity, userID \rightarrow User, type)

activityID, userID \rightarrow type

Event(eventID, name, description, occurrenceDate, creatorID \rightarrow User)

eventID \rightarrow name, description, occurrenceDate, creatorID

EventParticipant(participantID \rightarrow User, eventID \rightarrow Event)

4.2 Análise Forma Normal

De acordo com o modelo relacional apresentado, todas as relações respeitam a forma normal de Boyce-Codd e, conseqüentemente, a 3^a forma normal com a exceção da Relação User.

Em $\text{birthDate} \rightarrow \text{age}$ o elemento que se encontra do lado esquerdo da relação (birthDate) não é uma chave, constituindo deste modo uma violação à BCNF. No entanto, o lado direito da relação apenas é constituído por atributos primos fazendo com que respeite a 3^a forma normal, a qual exige que os atributos do lado esquerdo sejam chaves ou que o lado direito da relação seja apenas constituído por atributos primos.

Uma possível solução para este problema seria decompor User do seguinte modo:

- $\text{User1}(\text{birthDate}, \text{age}) \text{ birthDate} \rightarrow \text{age}$
- $\text{User2}(\text{id}, \text{birthDate}, \text{phoneNumber}, \text{gender}, \text{address})$

Quanto às restantes relações, uma vez que do lado esquerdo das dependências funcionais se encontram chaves primárias é possível concluir que respeitam a forma normal de Boyce-Codd, e como a 3^a forma normal é um 'super set' desta última também a irão respeitar.

Chapter 5

Restrições

5.1 Publisher

- *publisherID* é a primary key (key restriction, PRIMARY KEY);
- *name* é o nome do Publicador e não pode ser nulo (NOT NULL).

5.2 User

- *userID* é primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY) e não pode ser igual a nenhum atributo *pageID* de *Page*.
- *phoneNumber* é único a para cada utilizador (UNIQUE);
- *gender* apenas pode ter os valores 'M' (masculino) e 'F' (feminino) (CHECK);
- *birthDate* não pode ter valores nulos (NOT NULL) e tem que originar uma idade superior a 13 anos (CHECK NOW() - birthDate / 365 > 13)

5.3 Page

- *pageID* é a primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY) e não pode ser igual a nenhum atributo *idUtilizador* de *Utilizador*.

5.4 ChatParticipant

- (*participantID*, *chatID*) é a primary key (key restriction, PRIMARY KEY) and is a foreign key (referential integrity, FOREIGN KEY).

5.5 Friendship

- (*senderID*, *receiverID*) é a primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY) e têm que ser distintos;
- *date* corresponde à data do envio do pedido de amizade CHECK($\text{julianday}(\text{date}) \leq \text{julianday}(\text{'now'})$), tem de ser anterior ao momento atual e não pode ter valor nulo (NOT NULL);
- *state* corresponde ao estado atual do pedido de amizade e apenas pode possuir os valores 1 (aceite), 2 (pendente) e 3 (rejeitada) (CHECK).

5.6 EventParticipant

- (*participantID*, *eventID*) é a primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY).

5.7 PageFollower

- (*followerID*, *pageID*) é a primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY).

5.8 GroupMember

- (*memberID*, *groupID*) é a primary key (key restriction, PRIMARY KEY), é foreign key (referential integrity, FOREIGN KEY).

5.9 Group

- *groupID* é a primary key (key restriction, PRIMARY KEY);
- *name* corresponde ao nome do grupo e não pode ter valor nulo (NOT NULL);
- *adminID* é foreign key (referential integrity, FOREIGN KEY).

5.10 Event

- *eventID* é a primary key (key restriction, PRIMARY KEY);
- *name* corresponde ao *name* do evento e não pode ter valor nulo (NOT NULL);
- descrição corresponde à descrição do evento e não pode ter valor nulo (NOT NULL);

- *occurrenceDate* corresponde à data em que o evento se realiza e não pode ter valor nulo (NOT NULL);
- *creatorID* é foreign key (referential integrity, FOREIGN KEY).

5.11 Multimedia

- *multimediaID* é primary key (key restriction, PRIMARY KEY);
- *size* só pode tomar valores inteiros positivos (CHECK);
- *type* pode tomar os seguintes valores: ".mp3", ".jpg", ".png", ".wav", ".mp4" ... (CHECK).

5.12 Audio

- *audioID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY) e não pode ser igual a qualquer idImagem de Imagem e qualquer idVideo de Video;
- *length* só pode tomar valores inteiros positivos (CHECK).

5.13 Image

- *imageID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY) e não pode ser igual a qualquer idAudio de Audio e qualquer idVideo de Video.

5.14 Video

- *videoID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY) e não pode ser igual a qualquer idImagem de Imagem e qualquer idAudio de Audio;
- *length* só pode tomar valores inteiros positivos (CHECK).

5.15 Reaction

- *activityID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY);
- *userID* é foreign key (referential integrity, FOREIGN KEY);

- *type* representa as formas que a reação pode tomar e pode ter os seguintes valores: 1, 2, 3, 4, 5, 6 (CHECK).

5.16 Comment

- *commentID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY);
- *authorID* é foreign key (referential integrity, FOREIGN KEY);
- *postID* é foreign key (referential integrity, FOREIGN KEY).

5.17 Post

- *postID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY);
- *publisherID*, *multimediaID*, *pageID* e *groupID* são foreign keys (referential integrity, FOREIGN KEY).

5.18 Message

- *messageID* é primary key (key restriction, PRIMARY KEY) e foreign key (referential integrity, FOREIGN KEY);
- *multimediaID*, *authorID* e *chatID* é foreign key (referential integrity, FOREIGN KEY);
- *dateSent* corresponde à data de envio de uma mensagem e não pode ter valor nulo (NOT NULL).

5.19 Activity

- *activityID* é primary key (key restriction, PRIMARY KEY);
- *date* corresponde à data em que a atividade foi realizada. (tem de ser antes do momento presente)

5.20 Chat

- *chatID* é primary key (key restriction, PRIMARY KEY);
- *name* corresponde ao nome da conversa e não pode ser nulo (NOT NULL).