

Technical University of Moldova
Engineering department S.A.

Report

№: 3

AMOO
Subject: SEQUENCE DIAGRAMS

Author:
Prof:

Terman Emil FAF161
R. Melnic

Chisinau 2017

1 Theory

1.1 Sequence diagrams

In UML, collaboration between objects is explored in the informative aspect of their communications. In other words, the objects that interact, do nothing but a simple exchange of messages between them. In UML there are 2 types of interaction charts:

1. the objects collaborate with each other in time and then the sequence diagram is used to present the temporal peculiarities;
2. the structural particularities of collaborations between objects (the exchange of messages in space) can be investigated;

1.2 Object Oriented Analysis

OOA is an iterative stage of analysis, which takes place during the software development life cycle, that aims to model the functional requirements of the software while remaining completely independent of any potential implementation requirements. OOA can be described in two parts:

1. **Conceptual Object Oriented Analysis** So, what does this concept mean? Once we have created the use-case diagrams, we should describe the functionality of our application, in other words: to create the conceptual model. The main idea of this model is to describe the relations between classes and objects. To perform the conceptual model we should:
 - identify the objects;
 - refine the objects (remove duplicates);
 - draw the objects;
 - identify the relations;
 - identify their roles;
2. **Technical Object Oriented Analysis** The process of planning a system of interacting objects for the purpose of solving a software problem.
It should answer the question of how the conceptual model of the app should work and what are its methods. For this purpose, an UML language is used, so that everyone can understand the main idea of the model.

2 Tasks

2.1 Subdiagrams

In **Figure 1** is explained how the process of registration looks like. The actor is an unregistered user, and he interacts only with the site. While the Facebook platform, has interactions with both user and the database. First, the user tries to submit some registration data. Then the platform validates the input and loads the *Terms of use* from the database. The user then accepts the terms and the platform validates the registration data on the server. After a successful validation, the user is then registered in the database as an unconfirmed user. After that, a message requesting email confirmation is sent back.

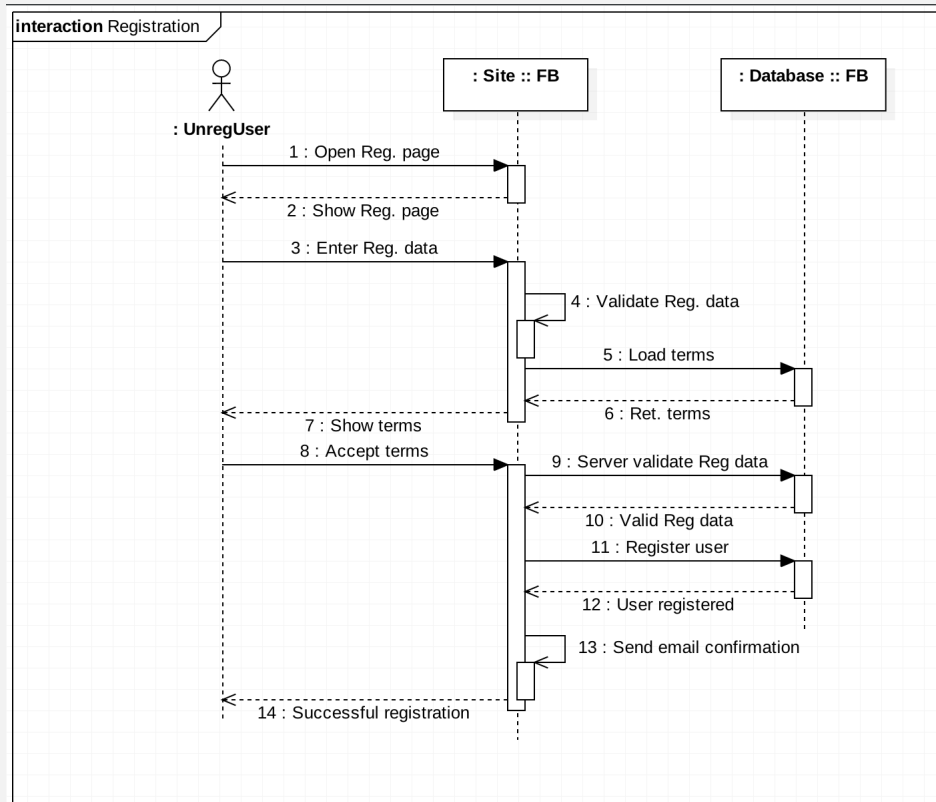
Moving on to user interaction, we have in **Figure 2** a friend request *sequence diagram*. In this case, two actors are involved: the sender (A) and the receiver (B). First, the sender tries to access the target. The platform then, returns a list of possible matches. After that, the first user submits a friend request. Then a validation is run and the request is registered on the database, followed by a notification to the *receiver*. Finally, the first user receives a *success* message.

Finally, in the last image (**Figure 3**), the diagram of creating an event is shown. First, the user starts with accessing the *Create event* page. Then, he submits the event's data, followed by a validation on the platform. The event is then created in the database. After that, the user, usually decides to invite people to his event: invitations are sent and the requests are registered in the database.

2.2 Functional requirements

What will happen?

1. Register user on the platform;
2. Register a friend request;
3. Register an event;
4. Send email confirmation;
5. Find requested users;



Img 1: Seq Diag: Registration

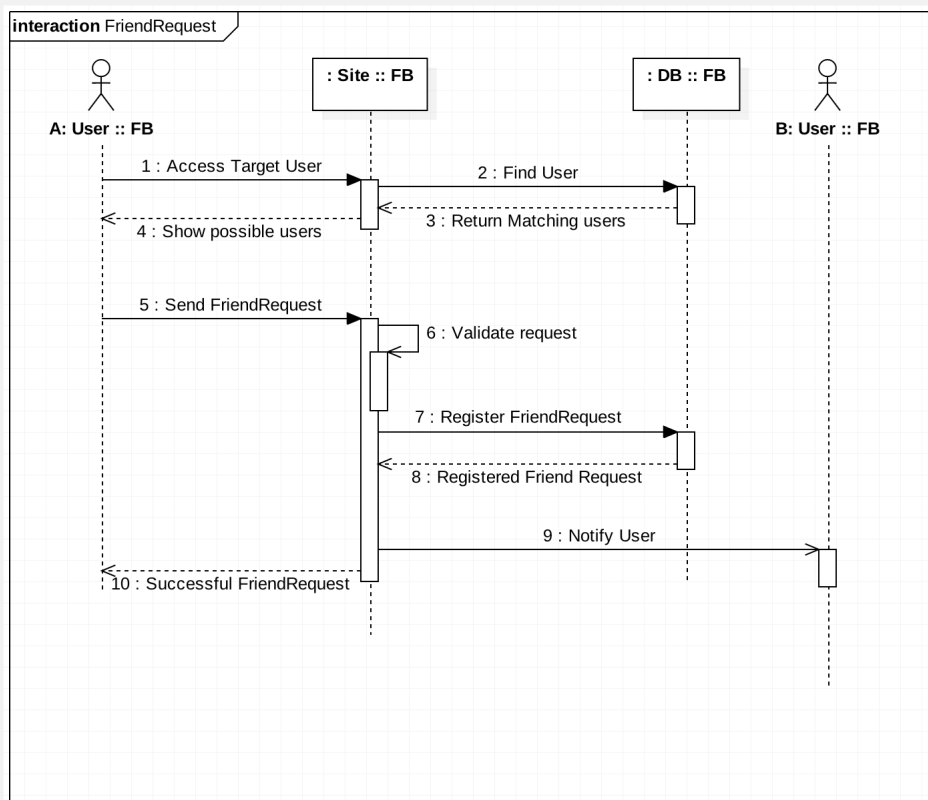
2.3 Non-functional requirements

How will it happen?

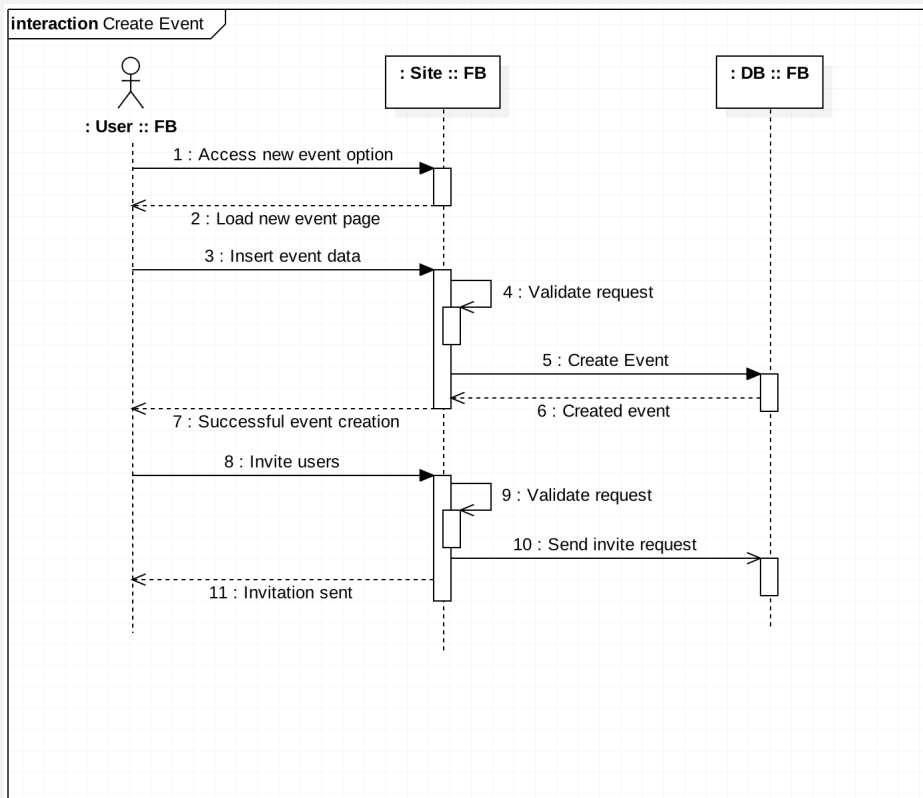
1. Only registered users can send friend requests;
2. Users can cancel the friend request, until it is accepted;
3. Users can receive notifications;
4. An event invite can be set to be public or private;
5. Event invites can be canceled;

3 Conclusion

In this laboratory work, I studied how to use sequence diagrams. I've learned how to properly use synchronous, asynchronous and return calls. I've also accumulated practical skills of developing time based diagrams.



Img 2: Seq Diag: Friend request



Img 3: Seq Diag: Create event