ComeTogether - an android app to meet new people

Christof Ochmann
University of Applied Sciences Zittau/Görlitz
02826 Görlitz, Germany
sichochm@hs-zigr.de

Ingo Körner
University of Applied Sciences Zittau/Görlitz
02826 Görlitz, Germany
siinkoer@hs-zigr.de

ABSTRACT

This work based on the document "Come-Together-App", which was created in Wirtschaftsinformatik II in a course of studies of tourism. The ideas of the project "Come-Together-App" are realized as a prototype in "ComeTogether - an android app to meet new people". Both, frontend and backend are analysed, desgined and implemented.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms, Design, Performance, Theory

Keywords

Parallel evolutionary algorithms, island model, spatial structures, offspring populations, runtime analysis

1. INTRODUCTION

In this project a prototyp named ComeTogether will be created. He runs on smartphones with the android operating system. ComeTogether is a mixture of an eventcalendar and a platonic touch. ComeTogether is an app to establish social contacts. It is no flirt or dating app, but an app to meet immediately a whole group of new people simultaneously and have fun with them. The core of the application consists of an offer function and a search function. With offer you can offer events with a participating group of people. With search you can find a group of people which participate in the event you have searched. Both, users who offer and users who search signalise with an offer or a search, that they want actively meet new people.

2. REQUIREMENTS ENGINEERING

Figure 1 shows the use case diagram for ComeTogether.

3. DESIGN

In figure 2 you can see the design for the UserService. UserServiceREST encapsulate the REST-Functionality of the UserService. UserPersistence access database with prepared statements p.e. to save, read or delete users. Between UserServiceREST and UserPersistence are service class and DAO class to encapsulate different levels of abstraction. Beside UserServiceREST there are the classes EventServiceREST, ParticipationServiceREST and MessageServiceREST. EventServiceREST in figure 3 creates, reads and deletes events. ParticipationServiceREST in figure 5 creates participations and gives back a list of participation for a given eventid or a given userid. MessageServiceREST in figure 4 creates, reads or deletes messages.

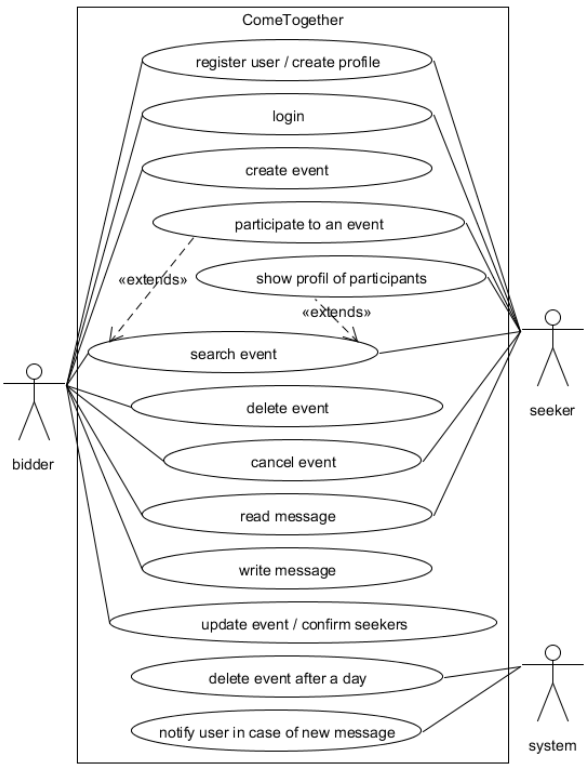

Figure 1: use case diagram

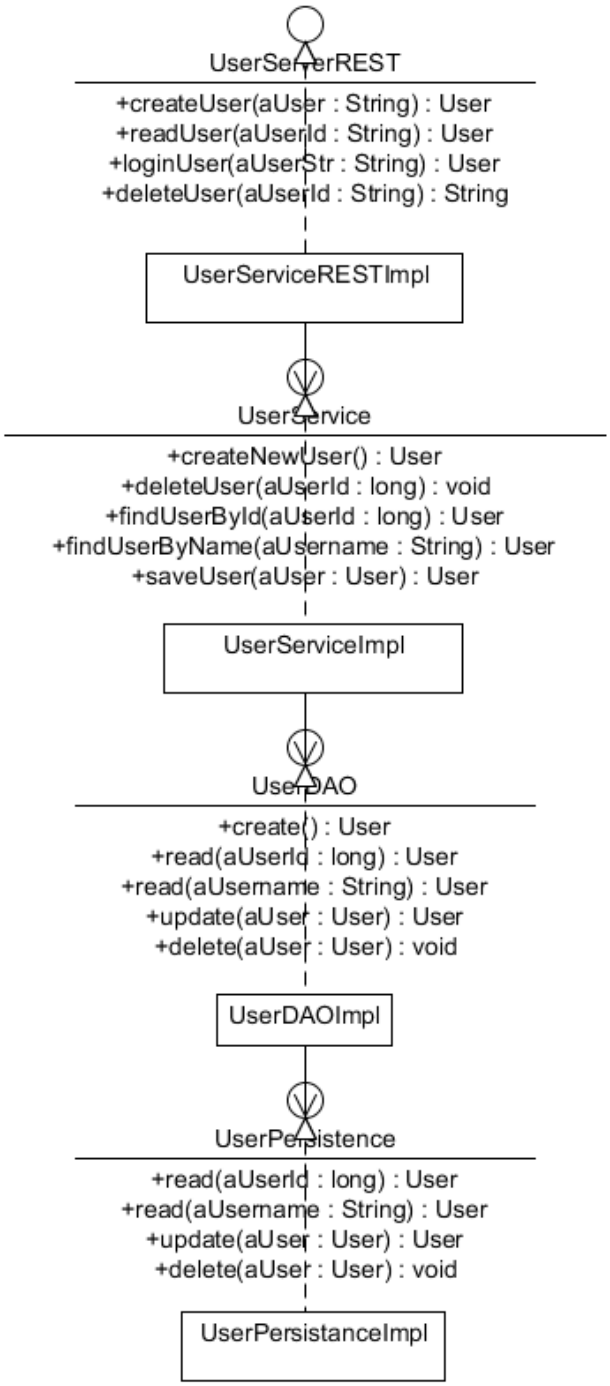

Figure 2: desing class diagramm UserServiceREST

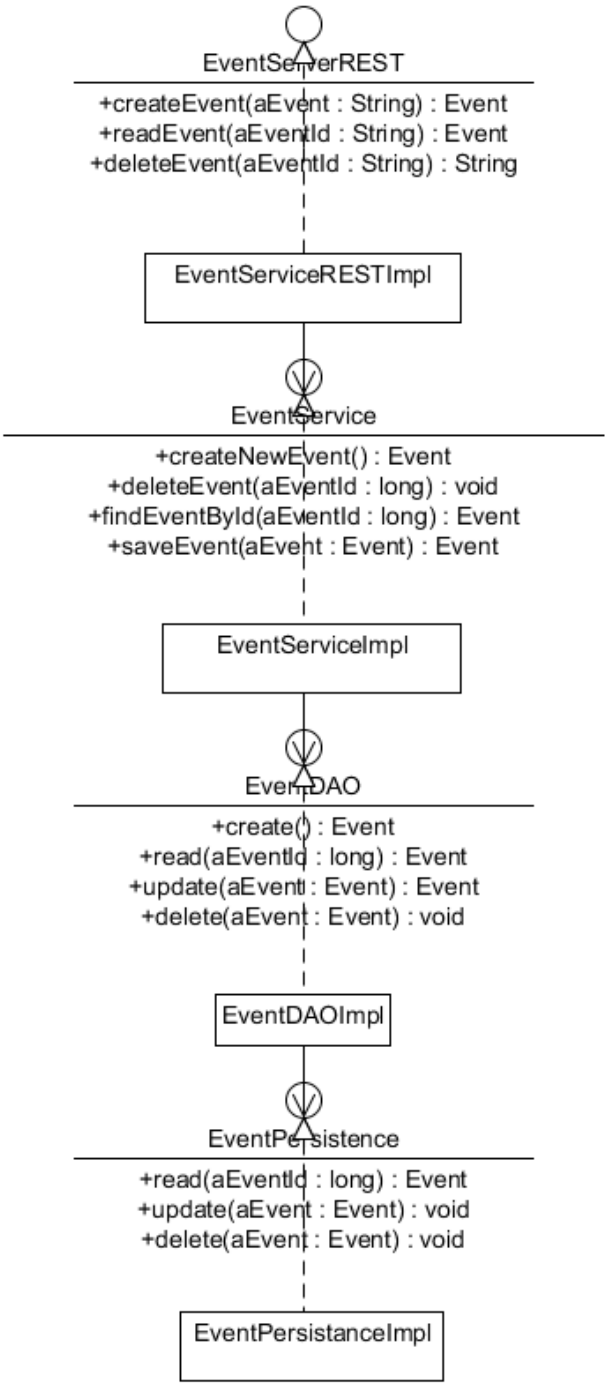

Figure 3: desing class diagramm EventServiceREST

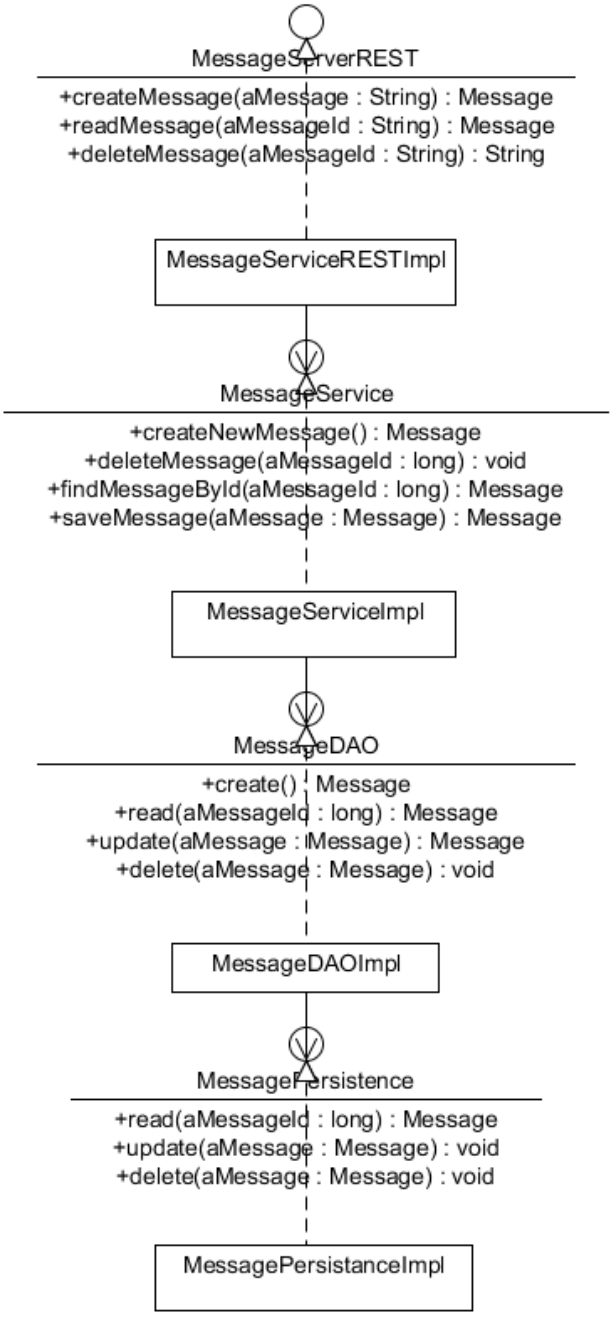

Figure 4: desing class diagramm MessageServiceREST

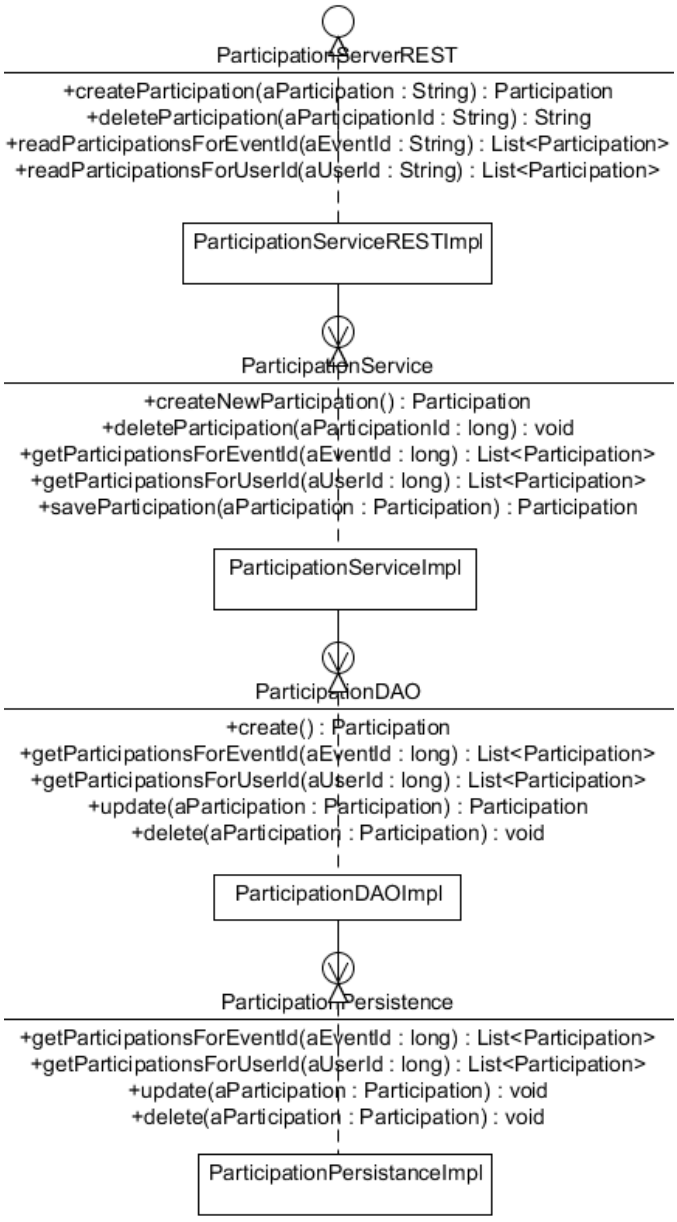

Figure 5: desing class diagramm ParticipationServiceREST

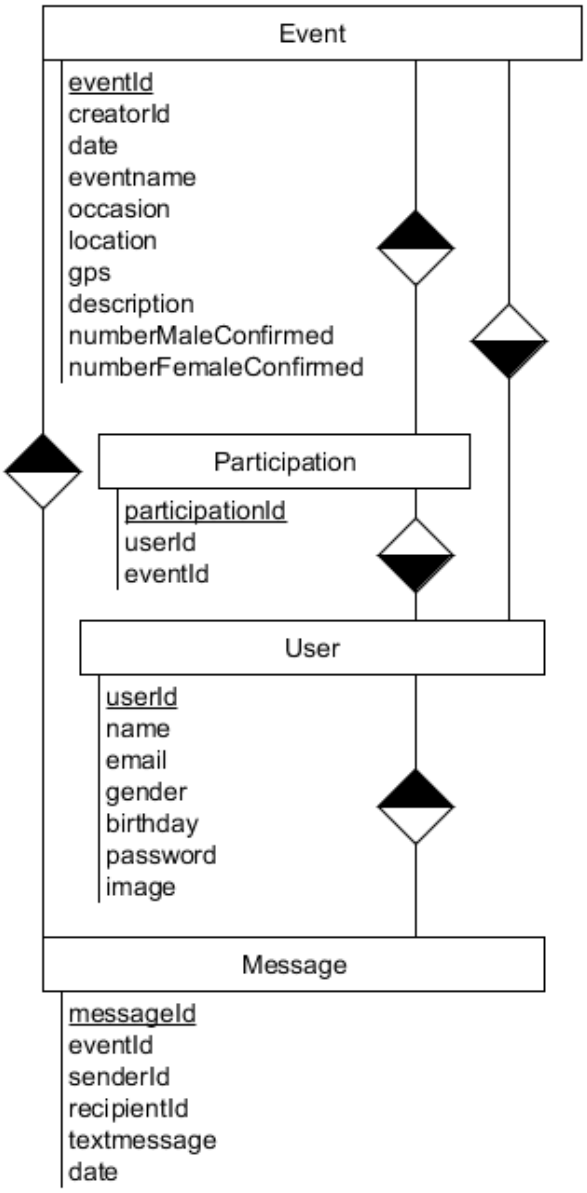

Figure 6: eer-diagram

4. DATABASE

In figure 6 you can see the data base design of ComeTogether.

5. PRELIMINARIES

6. PREVIOUS WORK

7. SORTING

...

8. SHORTEST PATHS

...

9. EULERIAN CYCLES

...

9.1 Edge Walks

...

9.2 Restricted Mutation Operators

...

9.3 Adjacency List Matchings

...

10. CONCLUSIONS

Acknowledgments

The authors would like to thank .....the German fast food industry for keeping us alive.

11. REFERENCES