

Exercise Sheet 4

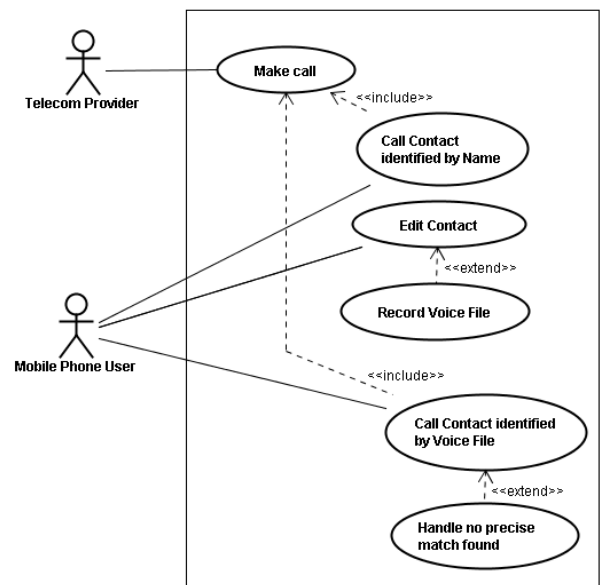
Due: Sunday 17.05.2009, 23:59:59 via SVN

For help, contact aosd-staff@lists.iai.uni-bonn.de (staff only) or
aosd-course@lists.iai.uni-bonn.de (staff and participants).

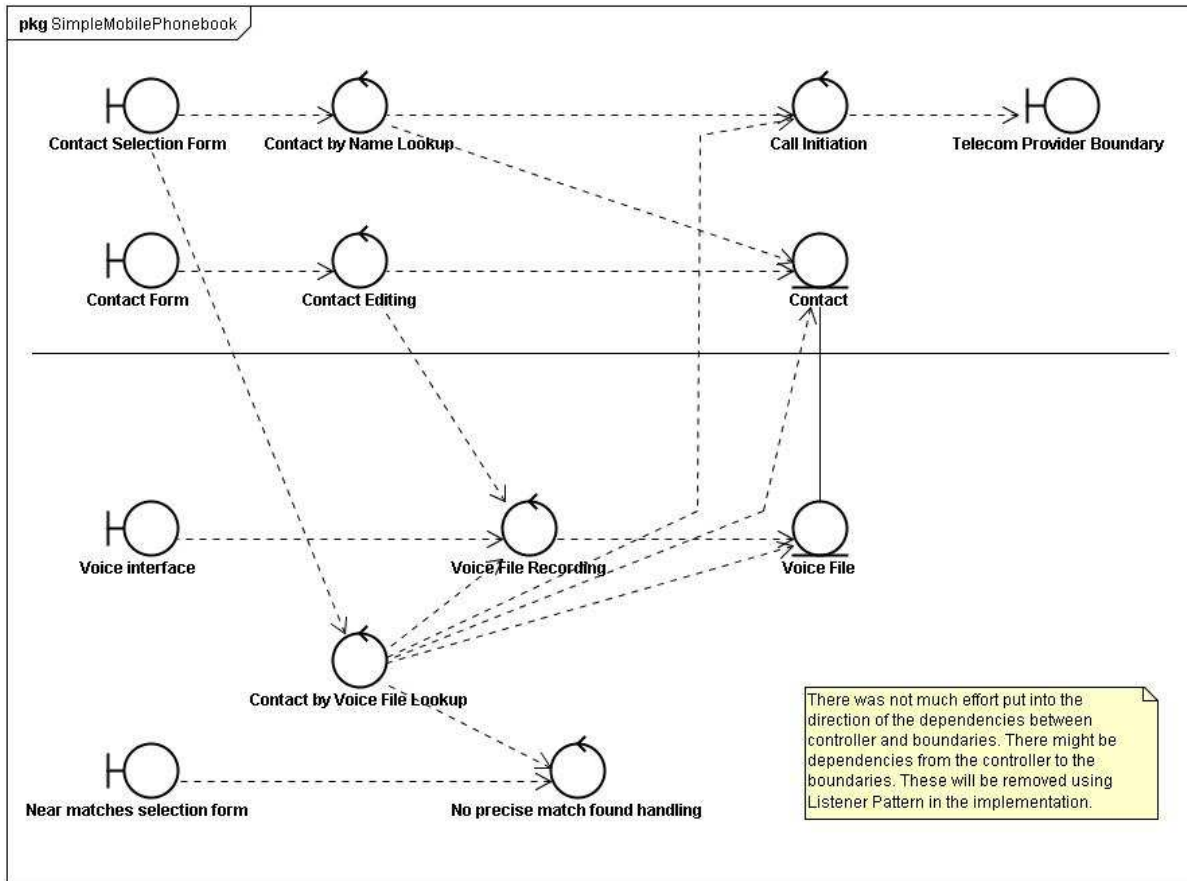
Please start working on the exercises early enough so that you can contact us in
time in case of problems. Don't expect us to be available during weekend!

Exercise 1: "UseCaseSlices" (5 Points)

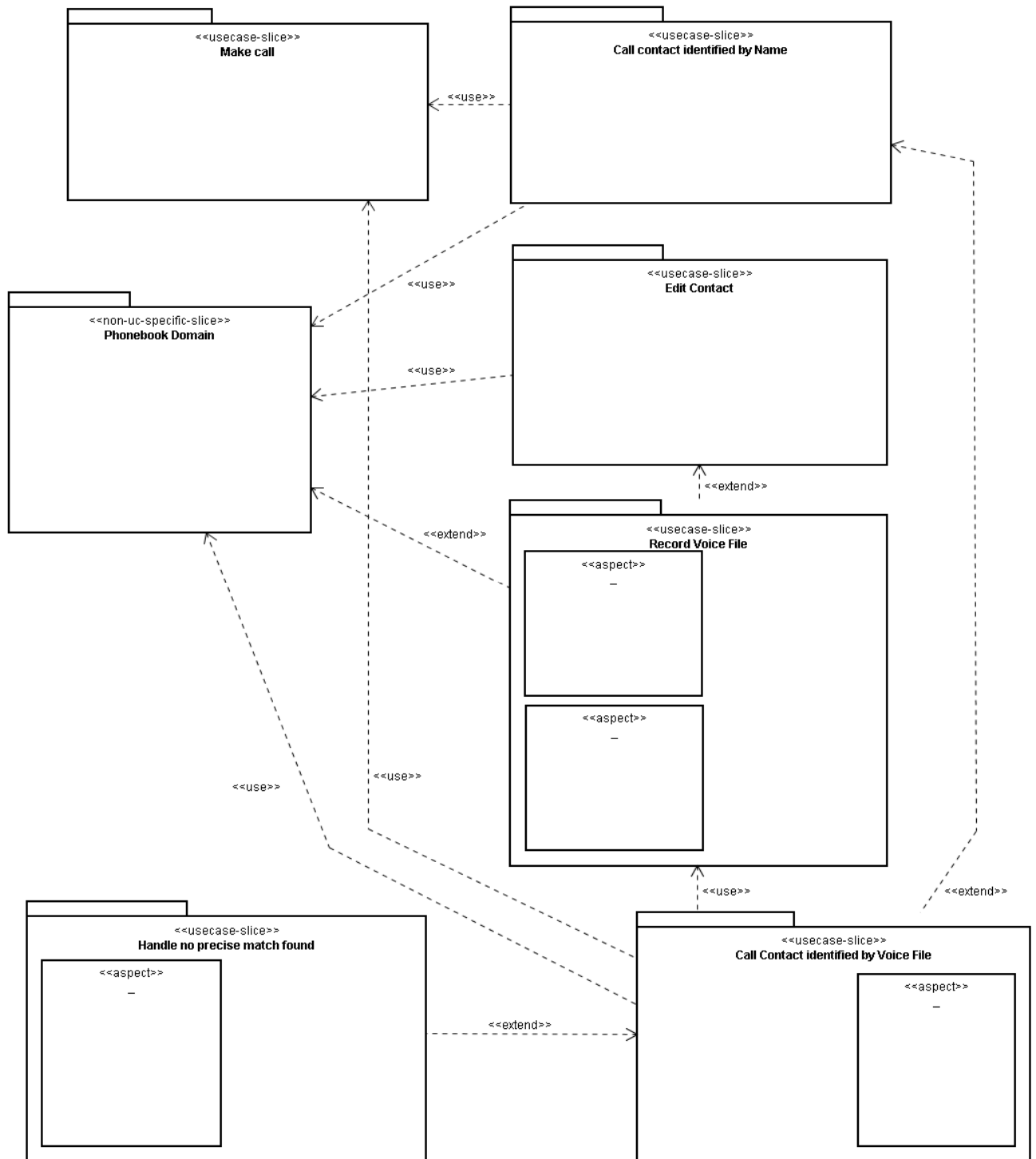
SimpleMobilePhonebook is a small application running on mobile devices. A user can edit the details of his contacts. A contact has a name, a phone number and an address. For every contact there can be a voice file recorded as an alternative mean to identify the contact. A user can make a call by entering the name of a contact or by voice lookup. To call a contact identified by voice file, he presses a special button on the screen; the device records his voice and compares it to the voice files for the contacts. If none of the voice files matches with a high enough score, a list with the best matches will be presented from which the user can select the contact. After successful selection of a contact the call is initiated.



From this Use Cases and the diagram you got the following Analysis Object Model Diagram:

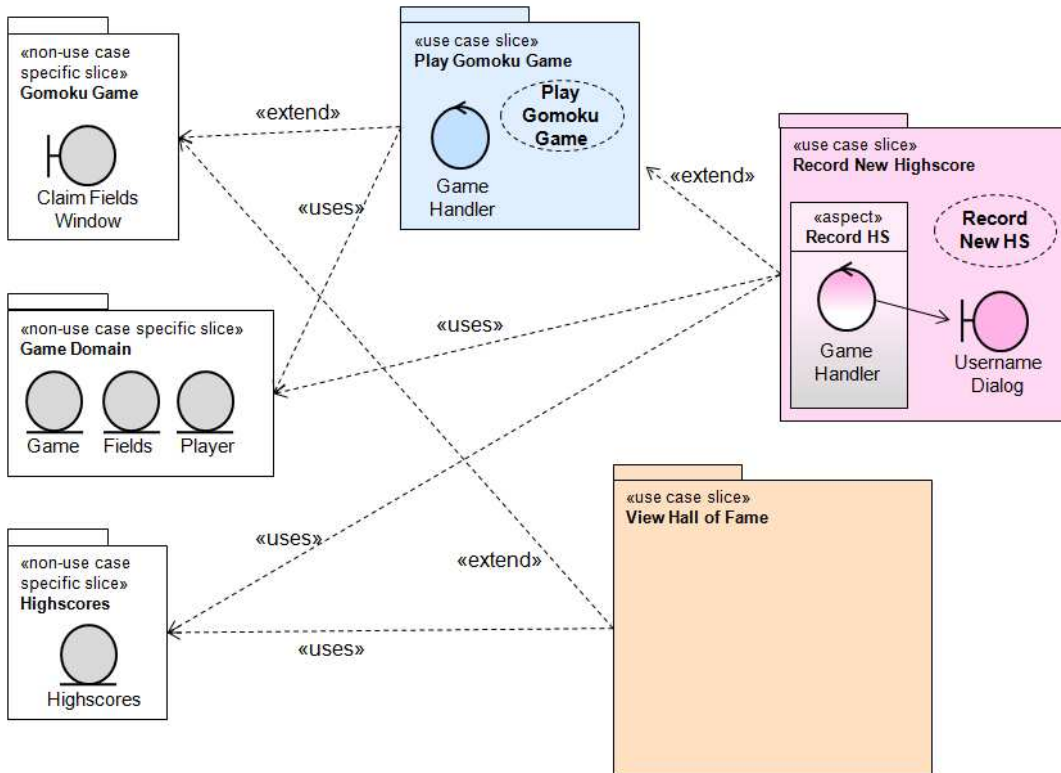
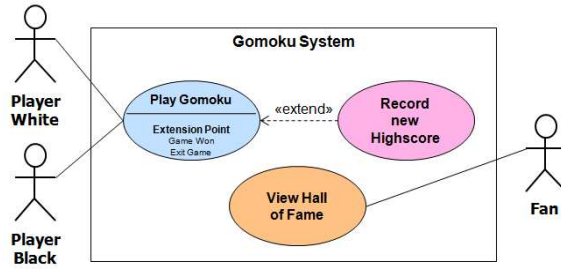


- There are several cases of “tangling” and “scattering” in this diagram. Which Objects are part of those cases?
- In the lecture you saw how to decompose such a diagram into use case slices. Fill in the objects into the use case slices we provided and add the needed aspects.
Do to this exercise you can use “jude” or any program you like as long as it produces a file that is readable from a Windows PC at room A-106. (jpg, bmp, pdf)

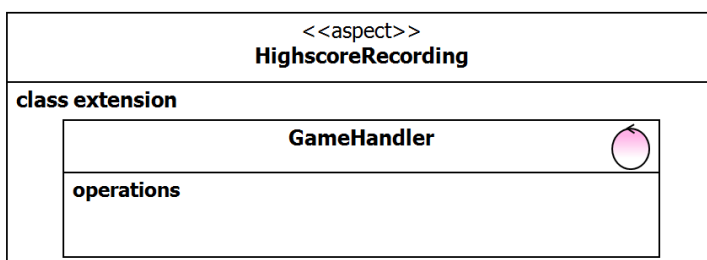
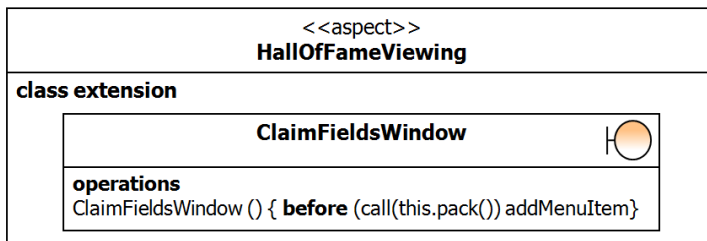


Exercise 2: "Gomoku" (6 Points)

We have implemented the Gomoku System that is described by the Use Case Diagram to the right. We used the use case driven analysis and produced the following Analysis Object Model.



- Fill in the missing analysis objects for the "View Hall of Fame". Be as detailed as we were in the other cases.
- Add the class extensions for "HighScoreRecording". The level of detail should be as high as in "HaloOfFameViewing":



- c) In your repository you will find the ES04_E02_GomokuApp application. The game is working correctly except that the “Hall of fame” is missing. We implemented most of the code except the aspects that you can find in the “control” package in the “record new highscore” folder and in the “ui” package of the “view hall of fame” folder. Please program the missing parts. *There are some minor changes you will have to make in your eclipse. The application won't work if you have the JDT weaving enabled. To disable it you have to go to your preferences, choose “JDT Weaving” and disable it. If it does not work automatically just follow the instructions.*

If you have any problems with that, just write us an email.

- d) You will also find an ES04_E02_GomokuClient and ES04_E02_GomokuServer project. This implements a distributed version of Gomoku. We implemented the server and now your job is to implement the aspects of the client. There are several hints in the code that will help you.