PS Software Engineering
December 19, 2001

ImmoWarrior

Report

Christian Koidl
christiankoidl@utanet.at

Academic Supervisor Mag. DI Dr. Roland Schwaiger

Department of Computer Science
University of Salzburg

Correspondence to:
Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Contents

Chapter 1

The Project

To learn the how-to's of "Software Engineering" we simulate a real life situation. A client, called e-Immo, comes to our software-development-firm and gives us an assignment.

Our Project is called ImmoWarrior.
The Task is to develope a computer based system, which collects and presents data about real estates.
The users of this system should of course be editors, who give the data input and readers. The readers could be real persons or foreign systems.
Now two possible scenarios are given:

- The collected data is forwarded to another system(using XML)
- The data is passed on to real persons. In this case the data has to be prepared for viewing.

As a special task the programm should be easy to be supported, means for instance no client - server model.
Further more e-Immo should have the possibility to influence the way in which the data is presented to the users.
In the first version of ImmoWarrior the essential functions should be included but with the option to be extended later on in cooperation with our group.
The expected deadline is January 2002.

Chapter 2

CRC Card Session

2.1 What is a CRC Card session?

CRC stands for Class, Responsibility and Collabotator. It is an object oriented analysis method to visualize the structure and behaviour of a system. Class means the object itself, responsibilities are the tasks the class has to do, and collaborators are other classes which are needed to fulfill the responsibilities. The found classes are written on standard index cards, that why this is called a CRC textitcard Session. The CRC Card Session should be performed in a group of 5 persons, to have the best productivity. The first part in a CRC Card Session is to extract possible classes form the problem domain. This could be done by taking all the nouns from the text as classes and all verbs as responsibilities. Afer that it has to be decided which classes and responsibilities are important for the project. This could be done in a brainstorming session. Not all classes and responsibilities have to be found and "useless" ones can also be included. The real set of cards will come out later on.

2.2 What we did on this topic

2.2.1 Our Job

Our Job was to inform us what a CRC Card Session is and how to perform it using the documents from the secretary and the internet. We had to prepare the session and go through the activities os the sesion with our group.

2.2.2 Preparation

First, as recommended, we gathered information about CRC Card Sessions, and after being well informed we atarted to walk through the problem domain. The extraction of Classes was not an easy task, because of the very loose problem domain. On this basis we started a brainstorming session to decide which classes are important and which have to be canceled. We also did some use case testing, not knowing that this would be the job for our next lesson. The result of our work were the following CRC cards:

Classname: **System**	
Responsibilities	Collaborators
interact with storage system forwarding information	Storage User

Classname: **USER**	
Responsibilities	Collaborators
read information write information	Real Estate System

Classname: **Storage**	
Responsibilities	Collaborators
store information read information	Real Estate System

Classname: **Real Estate**	
Responsibilities	Collaborators
keep information on re. read information	

For the 1^{st} meeting we also prepared some (many) questions on the problem domain.

2.2.3 The 1^{st} lesson

We started with giving our load of questions, what mainly lasted the whole lesson. What we found out was that our prepared cards were much to less. The additional information to our problem domain was:

- An access-permission system with the basic privileges read, write and administration should be supported
- For unregistered users, a little glimpse on the huge offer should be included.
- Users who want to see all offers have to pay.
- An automatic information system for the clients, giving them news about real estates, should be possible over e-Mail and WAP
- The login to our system via WAP and Internet is also needed.
- The selling of real estates is a task for the next version.
- Alltough the fact that a real estate can be sold meaning it should be removed from our database should be considered.
- The Text of a real estate should be supported with multimedia files
- A real estate search engine should be sported. enditemize With this new knowledge we added some CRC Cards:

Classname: **Admin**	
Responsibilities	Collaborators
access user interface view free offers log in configurate account create/modify account search for real estate view real estate info place/edit/delete real estate log off	user interface

Classname: **EditorAndUser**	
Responsibilities	Collaborators
access user interface view free offers log in configurate account search for real estate view real estate info place/edit/delete real estate log off	user interface

Classname: **Editor**	
Responsibilities	Collaborators
access user interface view free offers log in configurate account place/edit/delete real estate log off	user interface

Classname: **User**	
Responsibilities	Collaborators
access user interface view free offers log in configurate account search for real estate view real estate info log off	user interface

Classname: **Guest**	
Responsibilities	Collaborators
access user interface view free offers request new read account	user interface

Classname: **User Interface**	
Responsibilities	Collaborators
read input forward data show info start session	guest user editor editoranduser admin session

Classname: **Session**	
Responsibilities	Collaborators
create/modify/delete account initiate sending of account info forwards data to user interface initiate search close session access info storage request account info	user interface account send account info search for info info storage

Classname: **Account**	
Responsibilities	Collaborators
access info storage modify account info	session info storage

Classname: **Information Storage**	
Responsibilities	Collaborators
store account info store real estate info forward data	session account search for info

Classname: **Search For Info**	
Responsibilities	Collaborators
perform search forward search results	info storage session

In conclusion we found out, that thinking in object oriented manner is not too easy. But we were all confident that we'll have a lot of training the next 5 months.

Chapter 3

Essential Use Case

3.1 What is an Essential Use Case?

An Essential Use Case is a standard situation for our system. Such Essential Use Cases are used to find out the behaviour of the system. When playing through such a situation responsibilities and colaborators are found out and new classes could be added and others could be dropped.
How to do this use acse testing?
First you have to select a scenario. The CRC Cards now are distributed to the participating member. Each member has to concentrate on his card(s). As the scenario is started the members say what they (their classes) have to do an which other classes are needed to do the work. The new information is immediately written on the cards. This is how the responsibilities and Colaborators come out. If a card is not used in all tests, it can be dropped. When acting out a scenario the cards should be arranged in the way theiy work together to get a feeling for the system structure.

3.2 Our work on Essential Use Cases

3.2.1 Our Job

Our Job was to refine the poblem statement with the new knowledge of the last meeting and to have anothers CRC Card Session. In this second Use case Session Use Cases will come into play.

3.2.2 Preparation

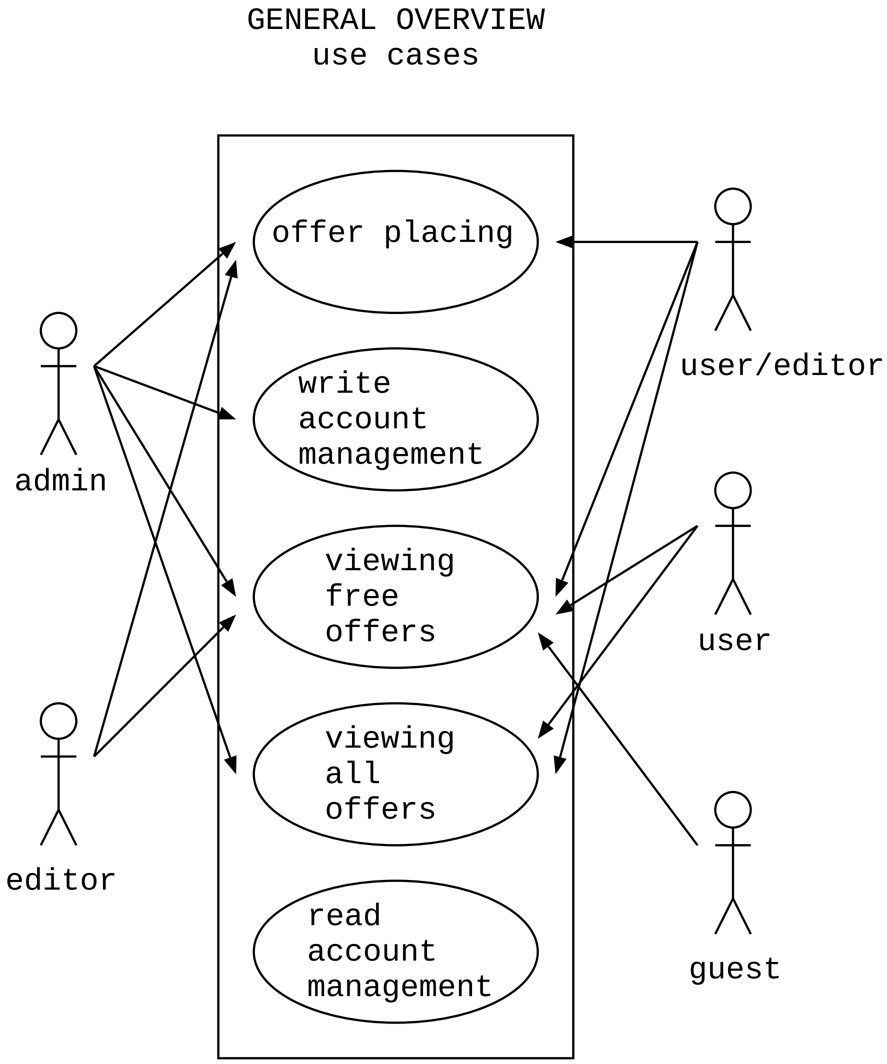
We started the preparation with playing through the old use cases, which we had from the last preparations. This was necessary because of the many new cards. The Use Cases we prepared were the following (Use Case diagrams in the next section):

Use Case Description

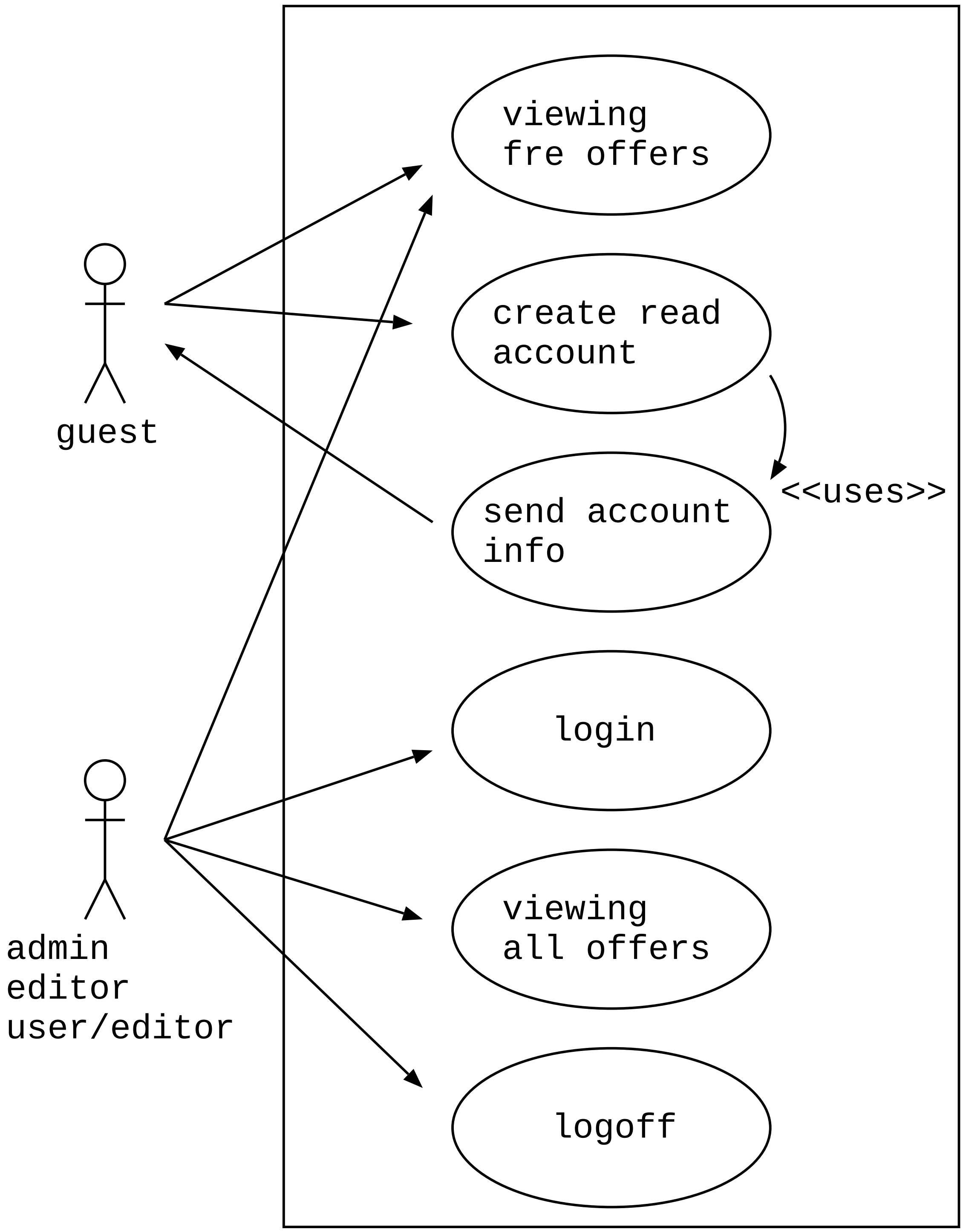
beginitemize

- view free offers Any user can view the free offers by visiting our homepage.
- create read account Any user can request read permission by submitting his personal data on the homepage. A confirm mail will be sent back from the system to the user including his username and password.
- view all offers Users who have read permission can view all offers by loggin in via homepage or WAP. Afterwards login he has the possibility to search for real estates with access to all real estates in the database.
- manage account Users who have already an account can make changes to their settings. Therefor the user has to log in and then is able to edit all his personal data, except his privileges.
- offer placing A user who got write privileges can add new real estates. In Order to do that he has to login and submit the real estate data. After he did the automatic forward function of the system is activated and the new data is sent to ll users who wish to get news about real estates.
- modify account This function is only accessable for administartors. An Administrator has to log in and then he can change data from any user including his privileges.
- create account This option is only accesssable for administrators too. An Administrator again has to log in and then he can enter the data from a user. When he submits the data username and password is sent to the user via e-Mail.

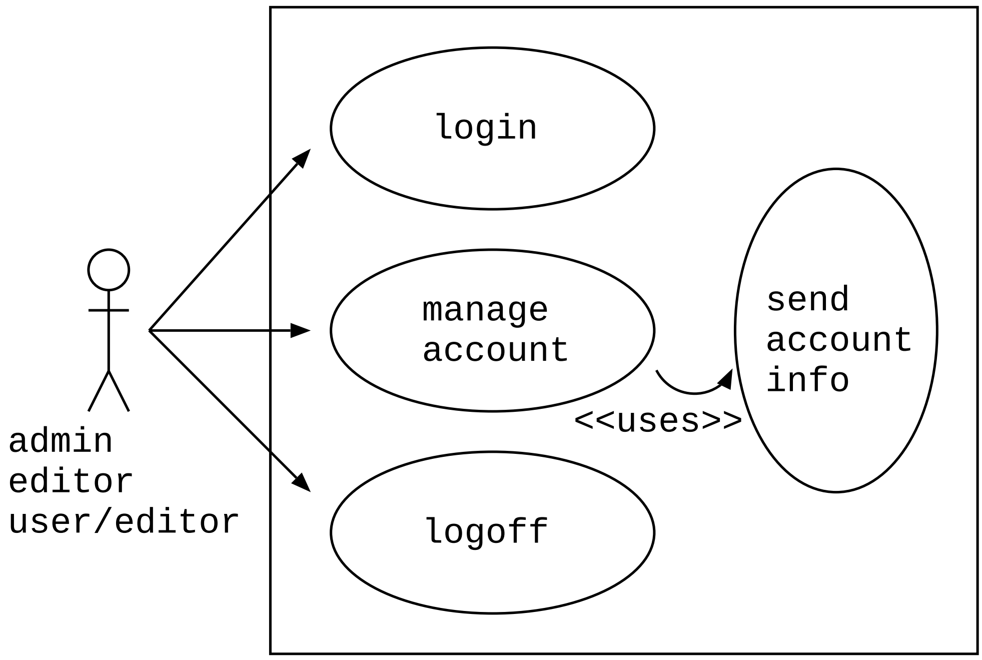
Use Case diagrams



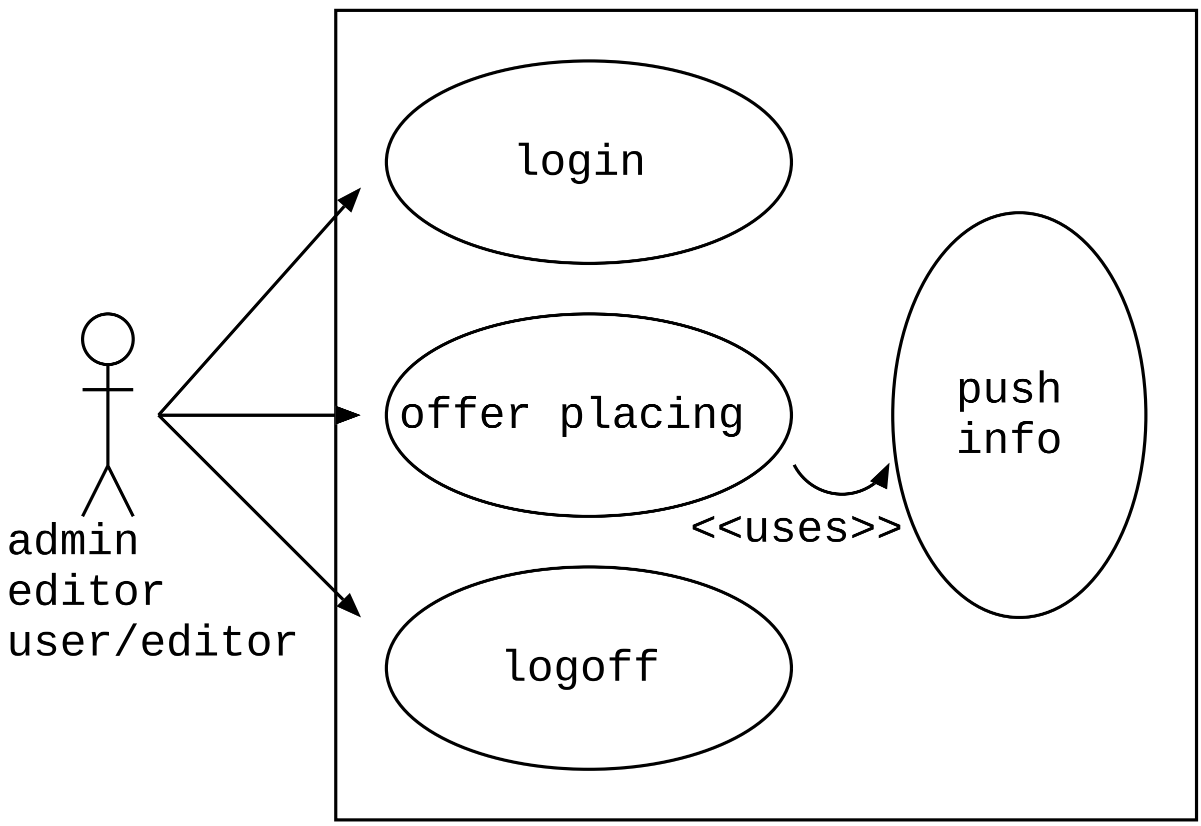
view free offers / create read account / view all offers



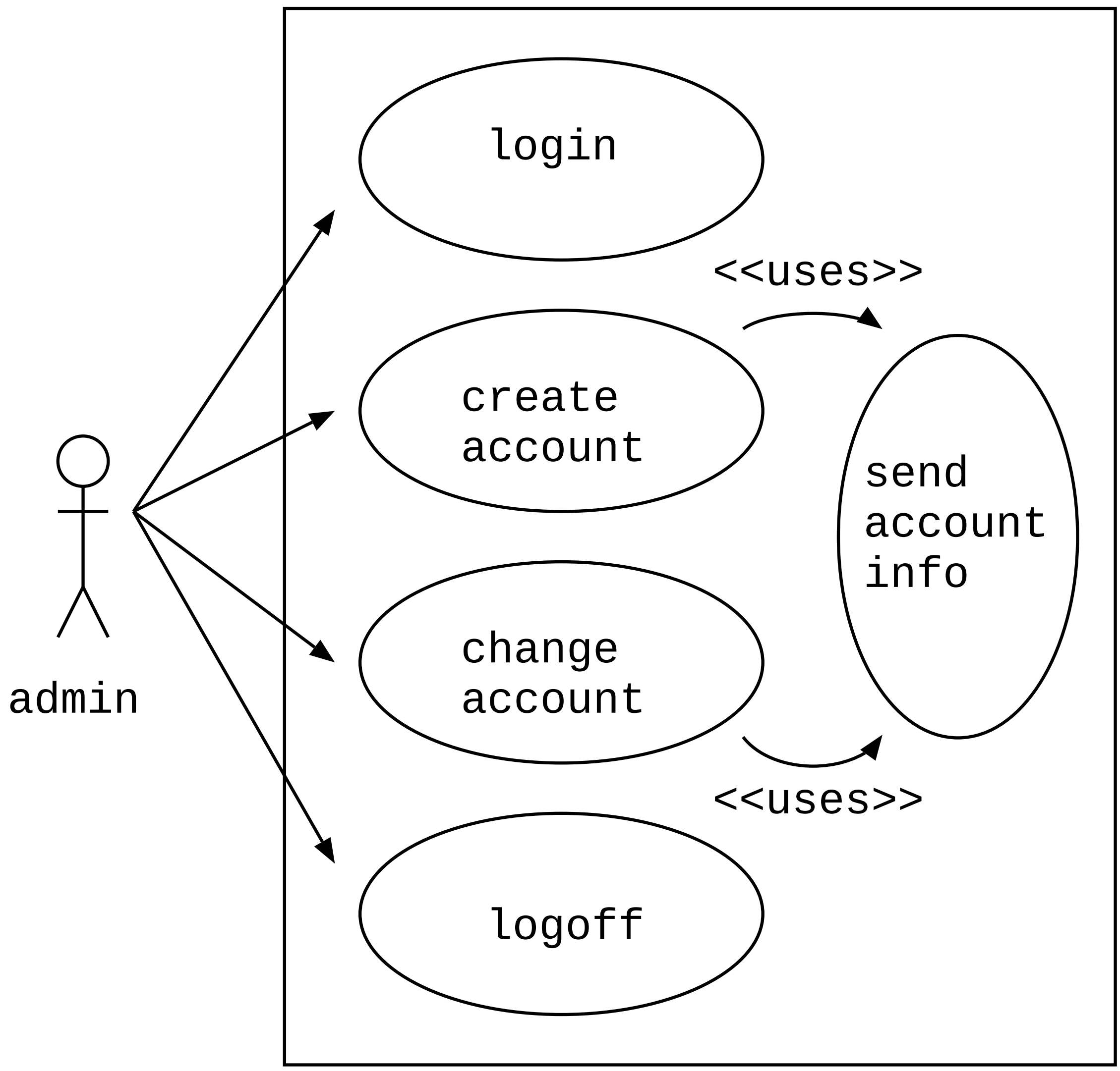
manage account



offer placing



create/change account



3.2.3 The 2^{nd} lesson

In the lesson we acted out two of the prepared Use Cases(because of the lack of time). These two were offerplacing and create read account. The CRC cards were distributed to the groupmembers and the development was:

create read account:

An actor visits the homepage. A session is started. The actor submits his data to the user interface. The user interface forwards the information to the session. The session checks if the data is correct, and if so passes the data on to the account. The account contacts the data storage which stores the data. The account returns a confirm message to the session. The session sends an e-Mail to the given address with the username and the password and alsogives an confirm message to the user interface. The user interface then closes the session.

offer placing:

An actor with write permission (editor, user-editor or admin) submits his login data to the user interface. The user interface starts a session and gives the data on. The session

requests the data for this actor from the account. The account gathers the requested data from the data storage and gives it back to the session. The session now checks if the username and password is correct, if it is correct and the actor has got a write permission the actor can place offers by submitting the data to the user interface. The real estate data is forwarded from the user interface to the session and the session saves the data in the data storage. When the data is stored the user interface gives a confirm mesage to the userinterface. When the actor now loggs out, the user interface closes the session.
Another task was to think about the Essential User Interface but there was no time to be dealt with it and this job was done in the next lesson.

Changes and new aspects:

- A new class has to be added: foreign system.
- The different actors should be developed through inheritance. The basic Actor is the guest and all other actors are derived from the guest where the different privileges are added.

Chapter 4

Essential User Interface

4.1 What is a Essential User Interface?

An Essential User Interface is the frontend of the System. This is what the user will see on his display and with what he directly interacts.

4.2 Our Essential User Interfaces

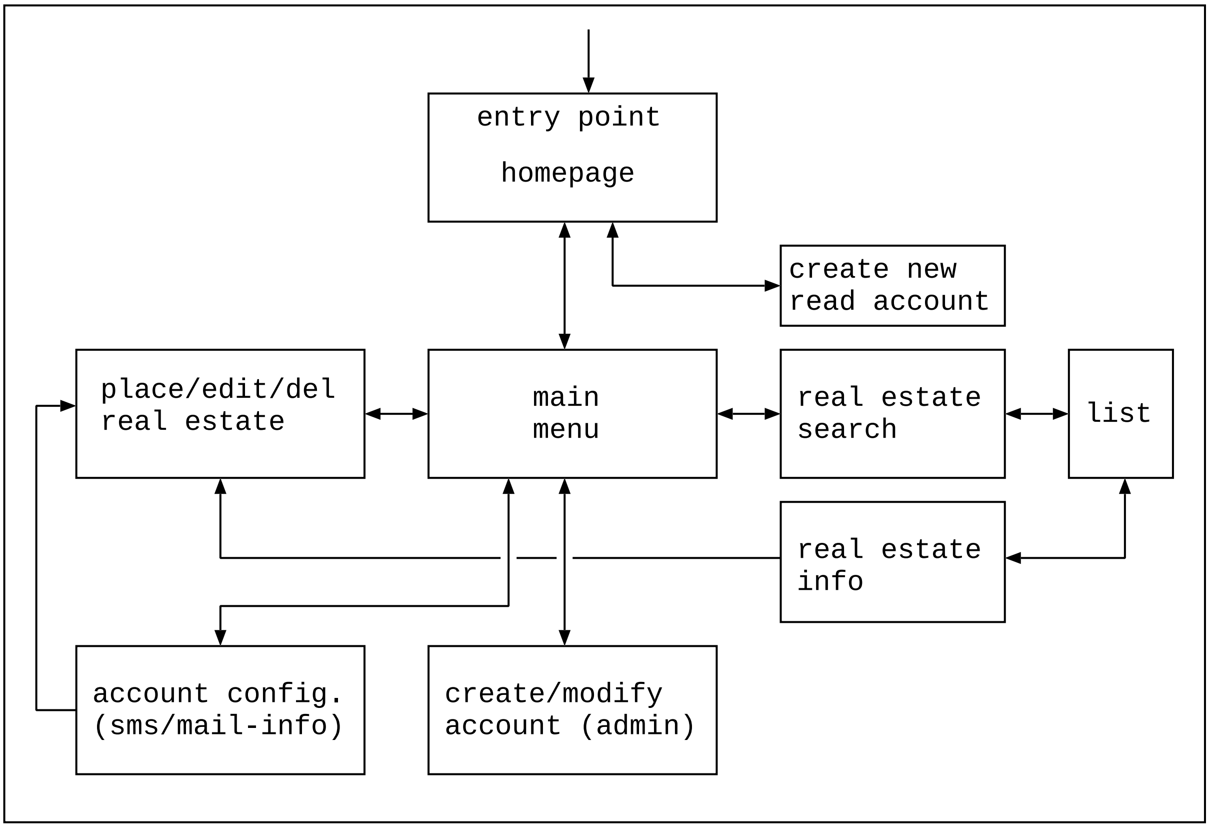
4.2.1 Our Job

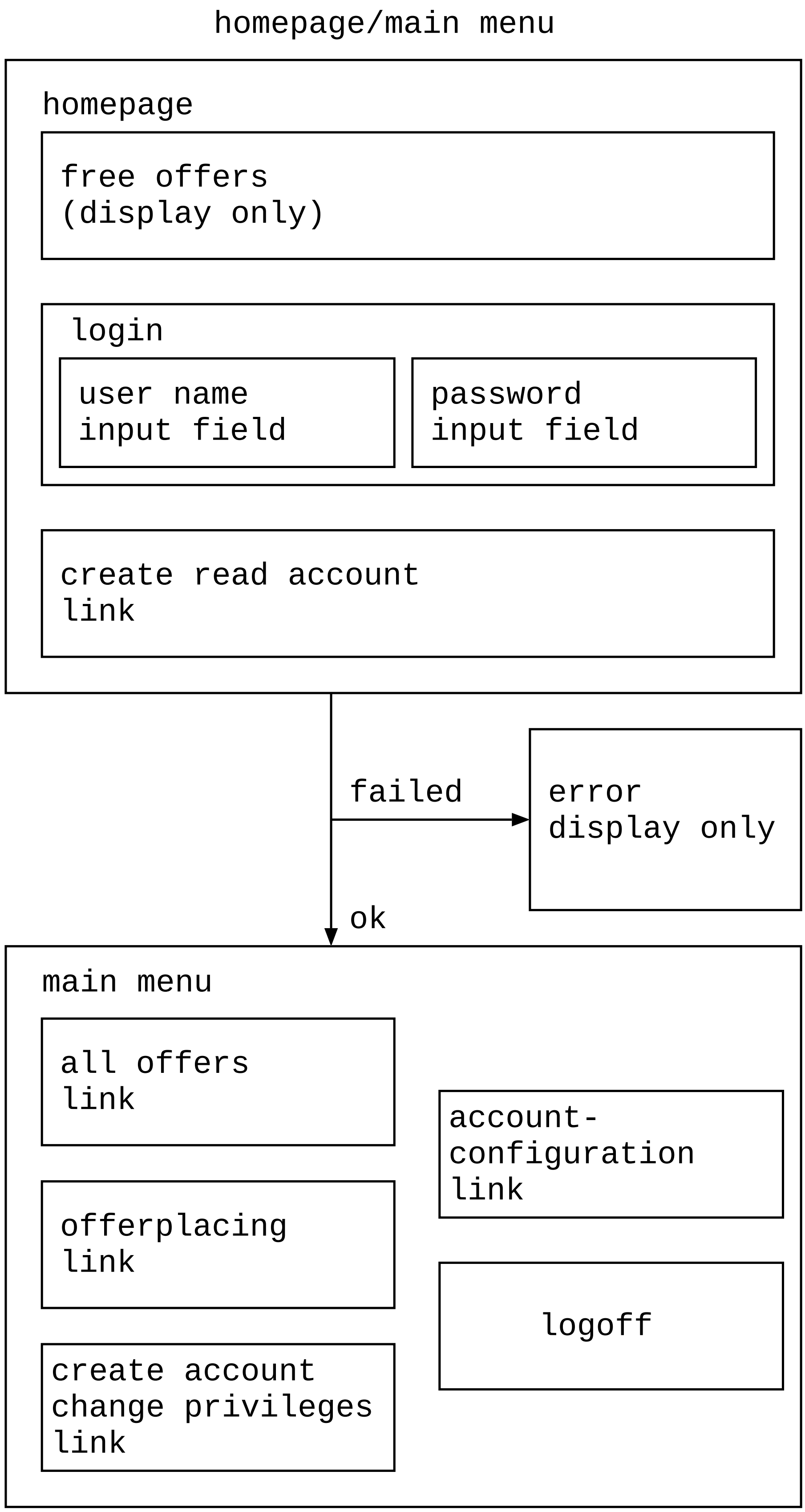
Our Job was to describe the Essential User Interface. The Essential User Interface together with the Essential Use Cases are the basis for our next Task the Class Diagram.

4.2.2 Preperation

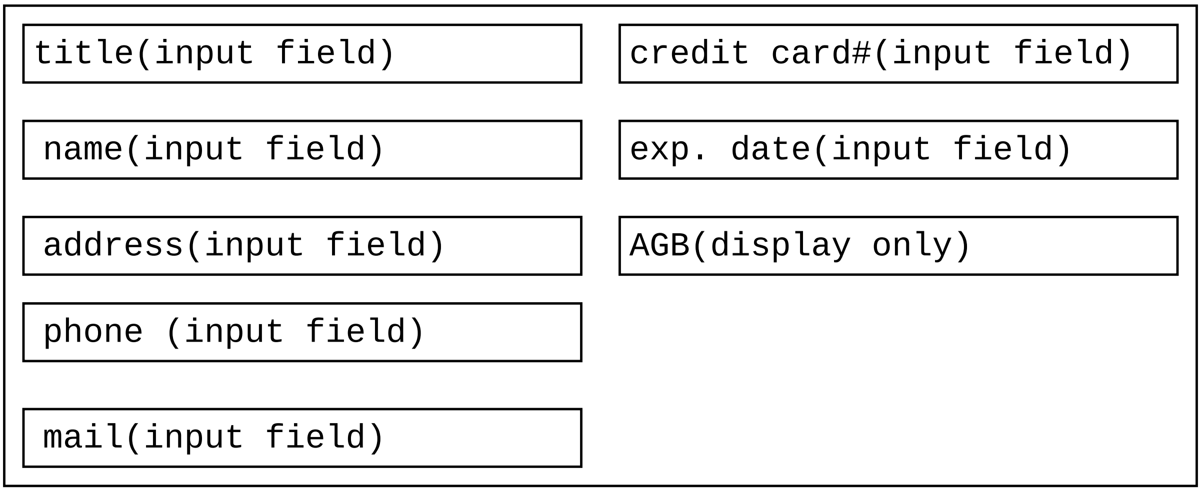
We developed Essential User Interfaces which fulfill all requirements, that were given at this time. Our User Interfaces looked like this:

OVERVIEW: User Interface

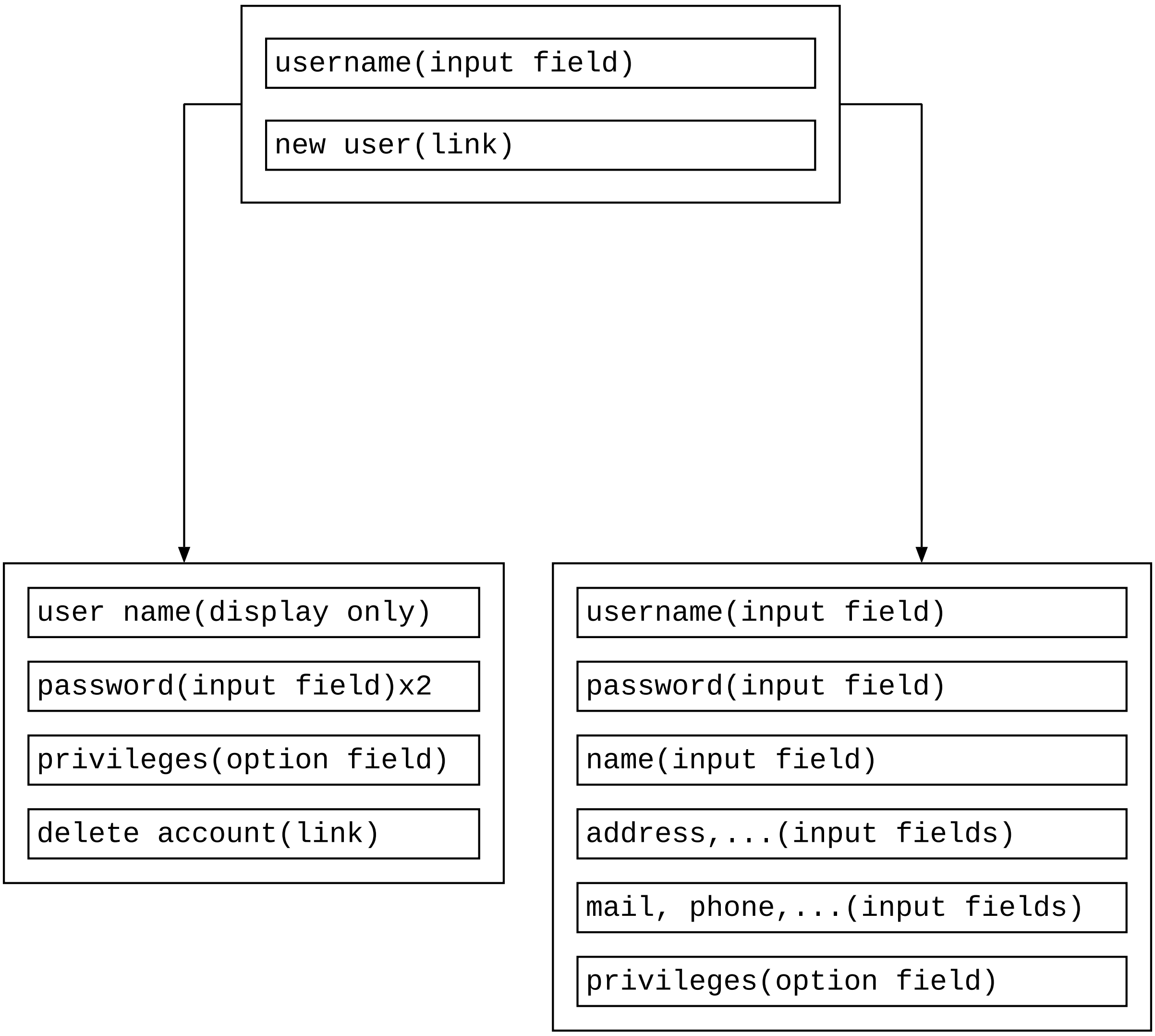



homepage/main menu
homepage
free offers
(display only)
login
user name
input field
password
input field
create read account
link
failed
error
display only
ok
main menu
all offers
link
account-
configuration
link
offerplacing
link
logoff
create account
change privileges
link

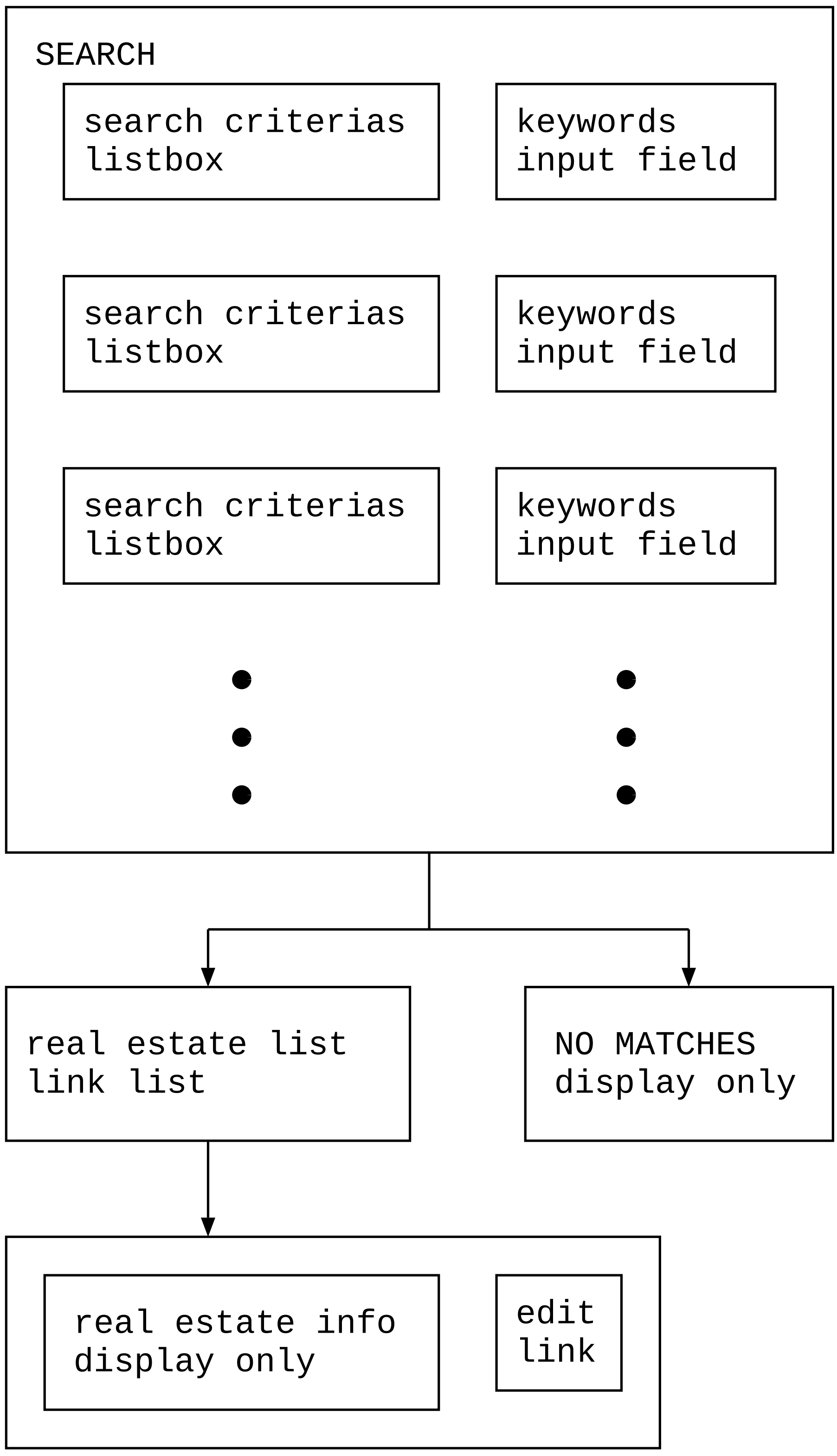
create read account



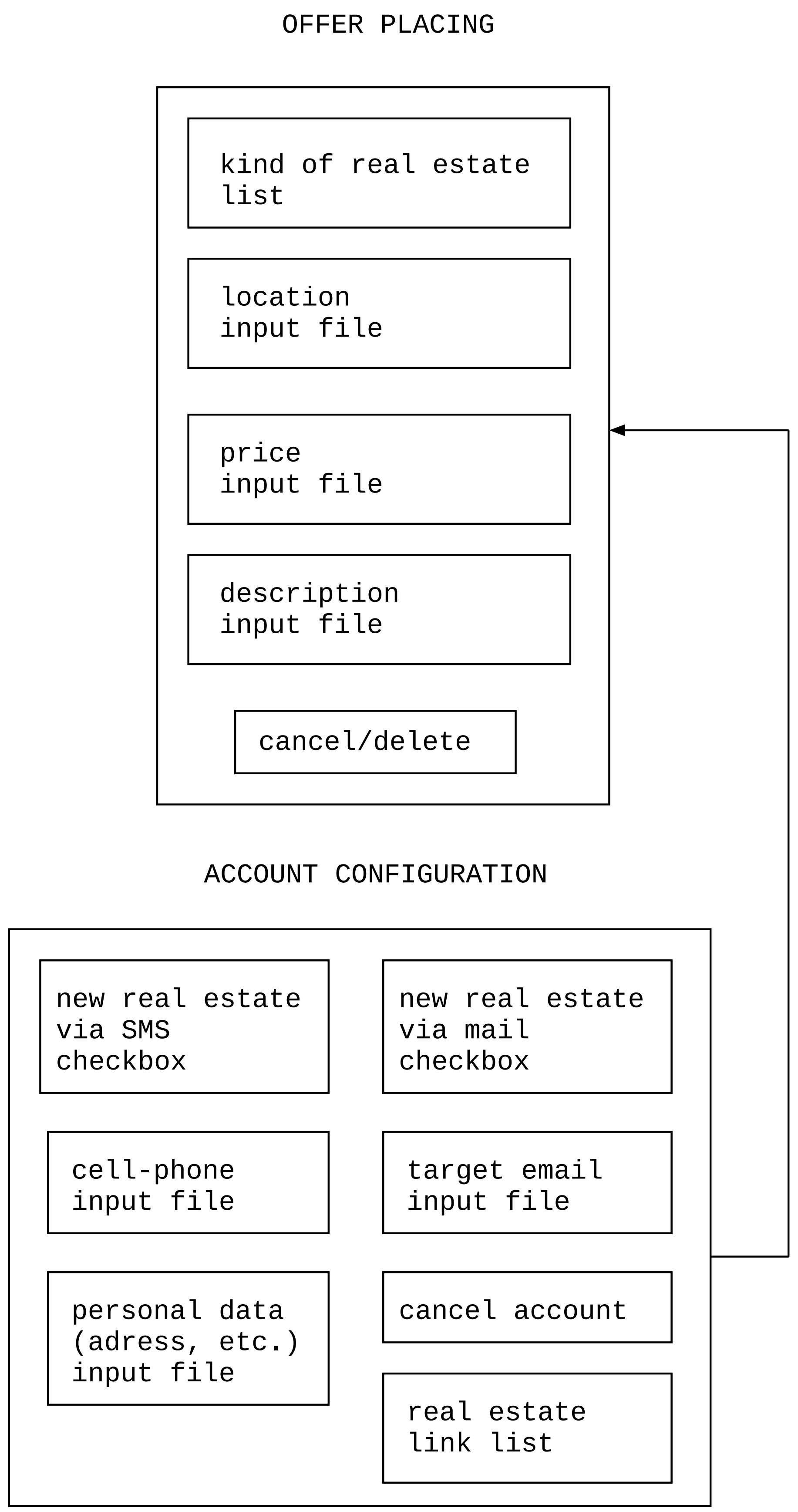
create account/change privileges



ALL OFFERS



SEARCH
search criterias listbox
keywords input field
search criterias listbox
keywords input field
search criterias listbox
keywords input field
real estate list link list
NO MATCHES display only
real estate info display only
edit link


OFFER PLACING
kind of real estate list
location input file
price input file
description input file
cancel/delete
ACCOUNT CONFIGURATION
new real estate via SMS checkbox
new real estate via mail checkbox
cell-phone input file
target email input file
personal data (adress, etc.) input file
cancel account
real estate link list

4.2.3 The 3^{rd} lesson

In the third lesson we made a presentation of our User Interfacesand following changes had to be done:

- The input fields for the personal data had to be altered: the address should be entered more exact and the name had to be split up.
- The Administration menus had also been improved by giving the administrator the posibility to search for members and then change their details.

Chapter 5

Class Diagrams

5.1 What is a Class Diagram?

Now we are going to make the step from the conceptual model to object oriented analysis. We have to make classes out of our CRC Cards. Therefore we have to take care of the class names (eg. no spaces) and also define the atributes and methods of the class. Inheritance, is also taken into account. Furthermore there are different relations between the classes: one to one, on to many, many to many (many may also mean zero). A Class Diagramm describes the static structure of the system.

5.2 Our Class Diagrams

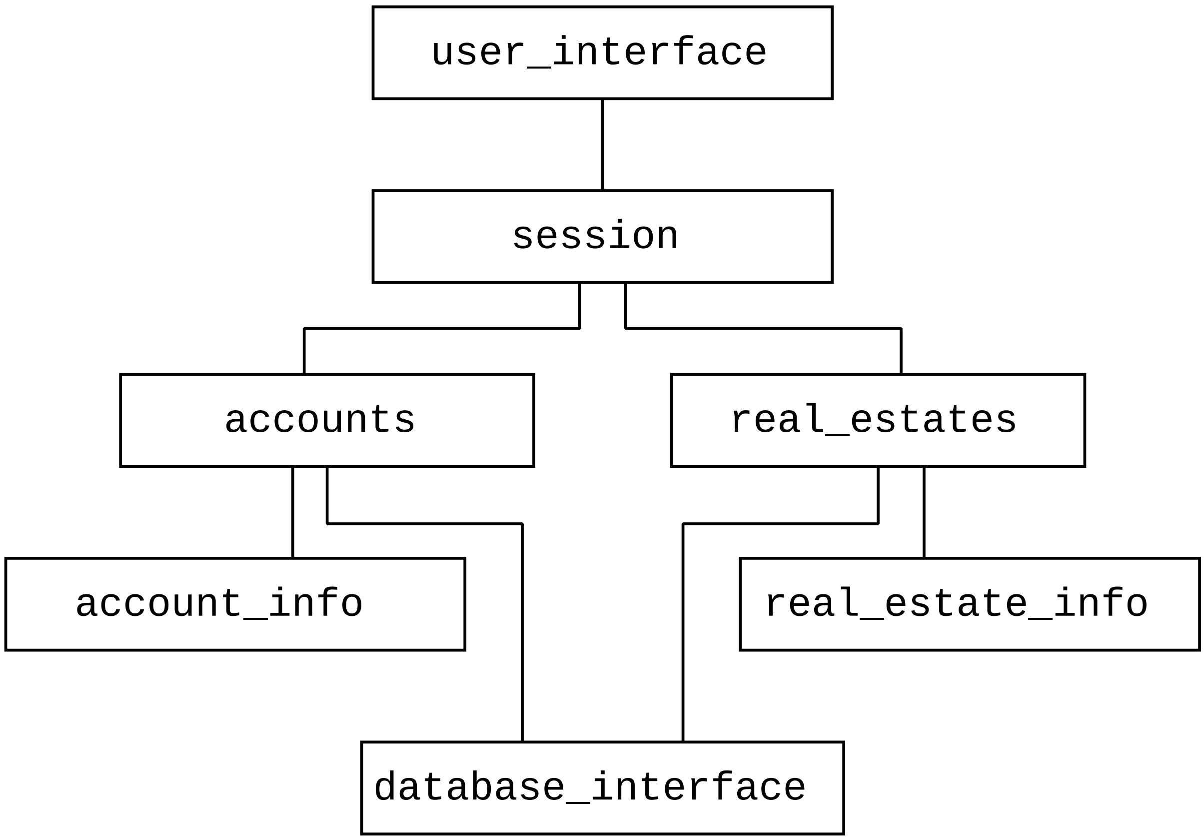
5.2.1 Our job

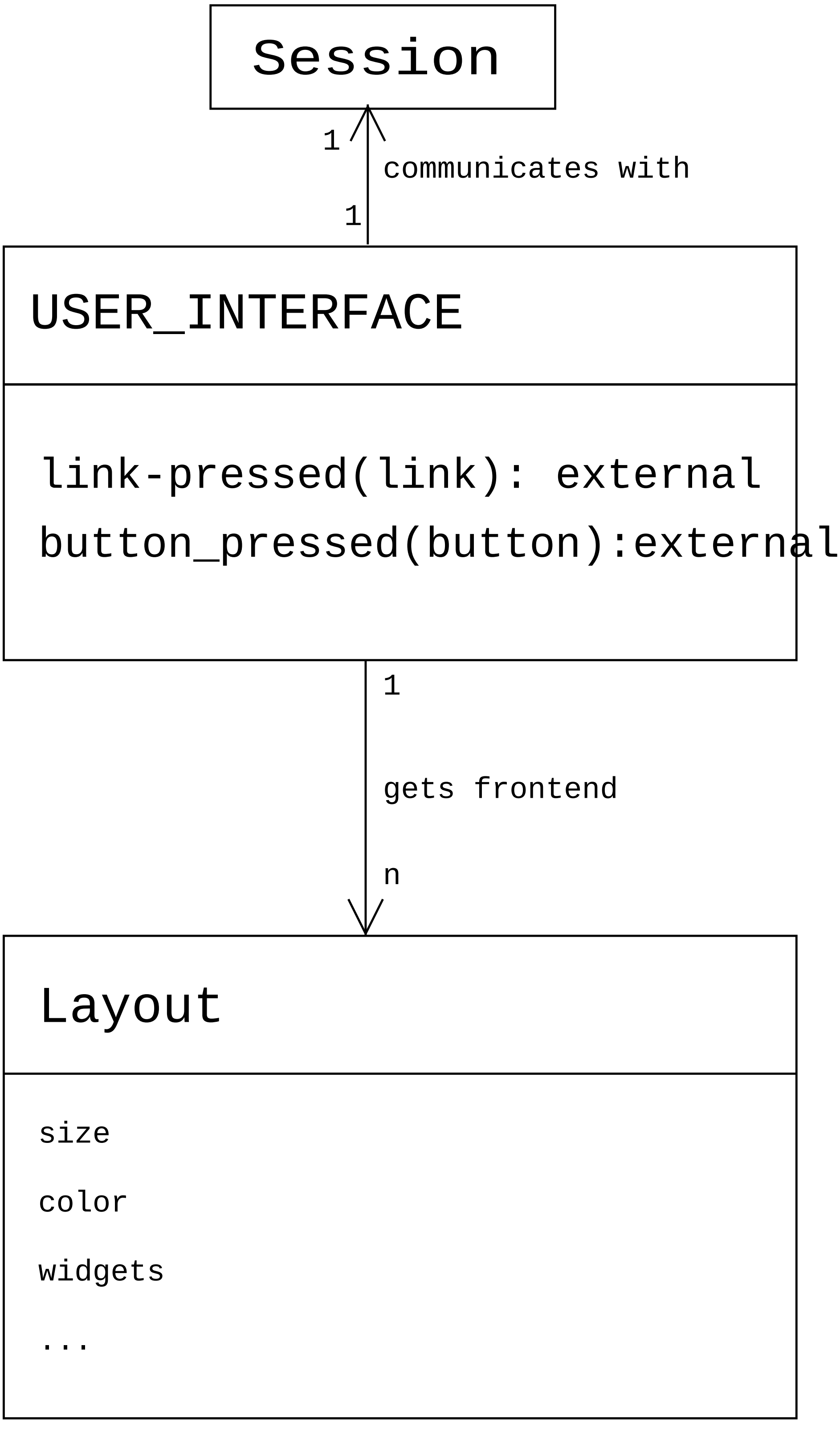
Our job was to develope Class Diagrams for our project.

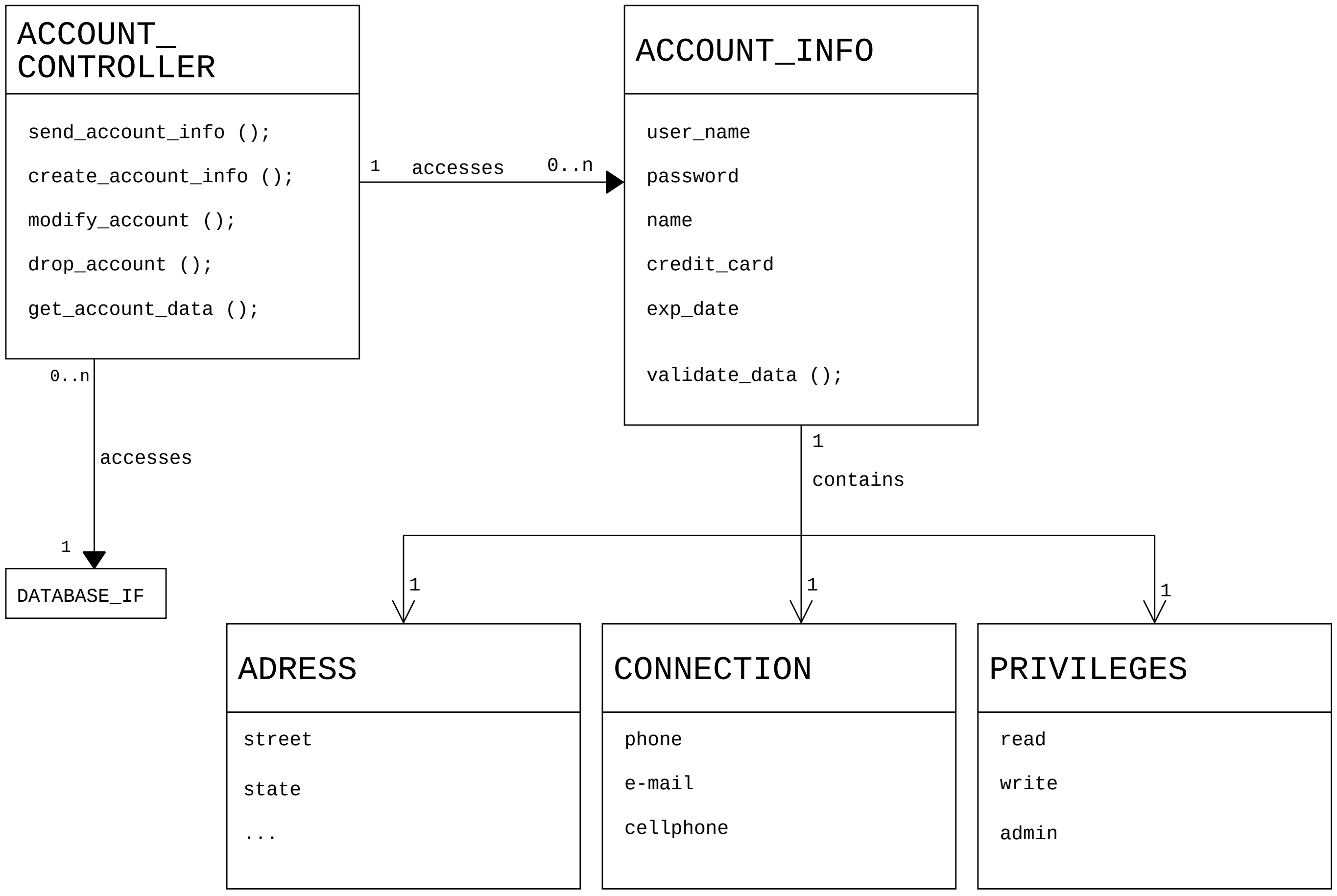
5.2.2 Preperation

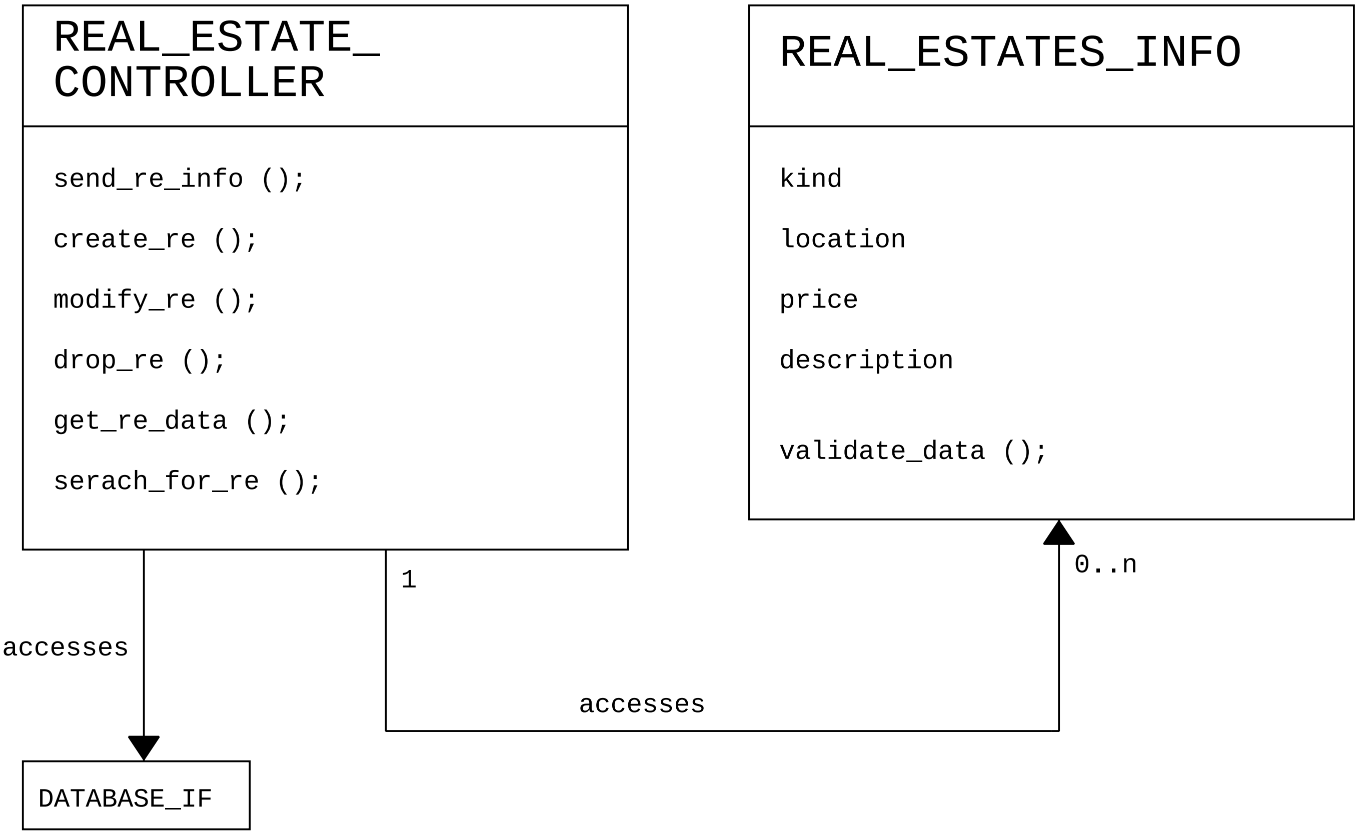
In our preperation we built the following Class diagrams:

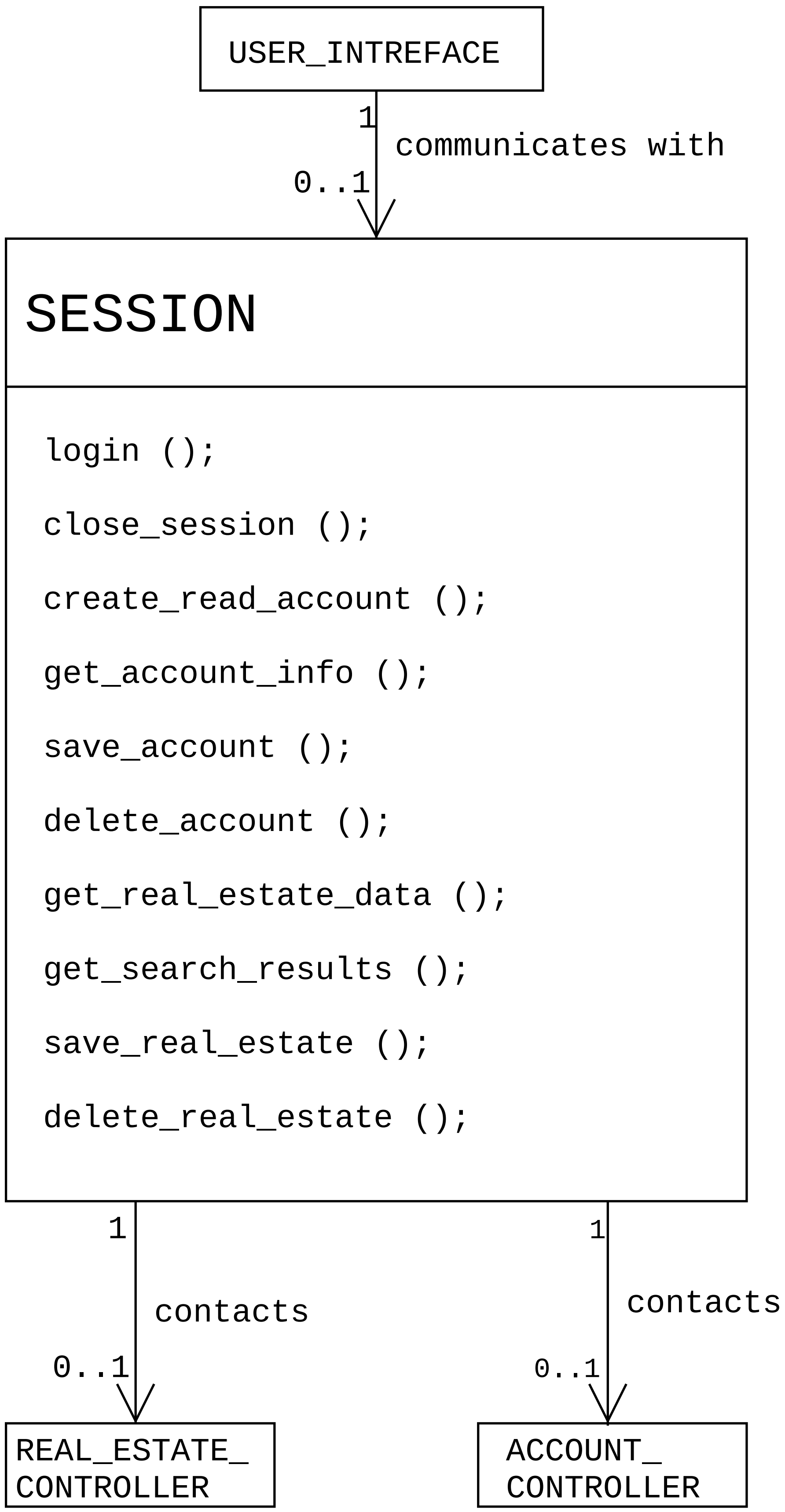
overview

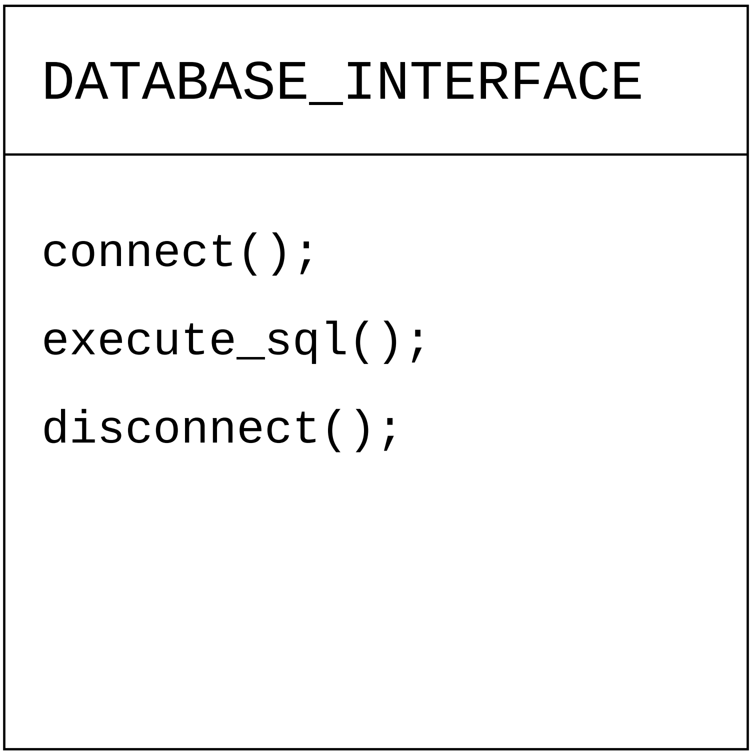

user_interface
session
accounts
real_estates
account_info
real_estate_info
database_interface


Session
1
communicates with
1
USER_INTERFACE
link-pressed(link): external
button_pressed(button):external
1
gets frontend
n
Layout
size
color
widgets
...


ACCOUNT_
CONTROLLER
send_account_info ();
create_account_info ();
modify_account ();
drop_account ();
get_account_data ();
1
accesses
0..n
ACCOUNT_INFO
user_name
password
name
credit_card
exp_date
validate_data ();
0..n
accesses
1
DATABASE_IF
1
contains
1
1
1
ADRESS
street
state
...
CONNECTION
phone
e-mail
cellphone
PRIVILEGES
read
write
admin


REAL_ESTATE_
CONTROLLER
send_re_info ();
create_re ();
modify_re ();
drop_re ();
get_re_data ();
serach_for_re ();
REAL_ESTATES_INFO
kind
location
price
description
validate_data ();
1
0..n
accesses
accesses
DATABASE_IF


USER_INTREFACE
1
communicates with
0..1
SESSION
login ();
close_session ();
create_read_account ();
get_account_info ();
save_account ();
delete_account ();
get_real_estate_data ();
get_search_results ();
save_real_estate ();
delete_real_estate ();
1
contacts
0..1
1
contacts
0..1
REAL_ESTATE_
CONTROLLER
ACCOUNT_
CONTROLLER



5.2.3 The 4^{th} lesson

In the meeting we again presented our diagrams and some changes had to be made:

- We had to create a new class called address which includes the atributes: country, state, zip, country, first name and last name.
- It also came out that some classes could be dropped. These were: homepage, interested_party, search_for_info and interrested_party.

Chapter 6

User-Interface Prototyping

6.1 Our Job

In this phase our job was to create a prototype of the user interfaces in WML (Wireless Markup Language) and HTML (Hypertext Markup Language). These Protoypes should not be presented with slides but with programs.

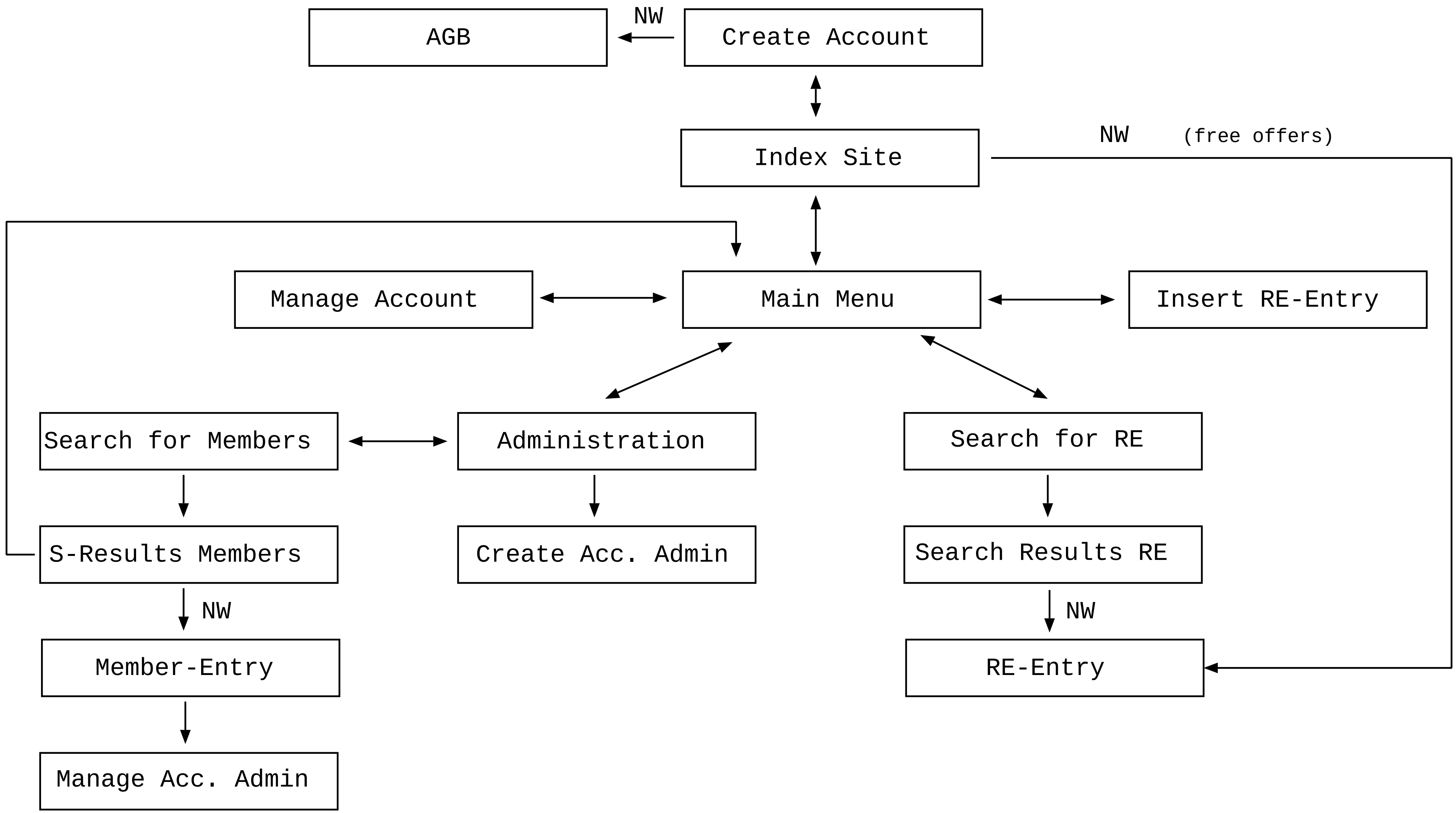
6.2 Preperation

The implementation of the HTML user interface was not too difficult, because we already had some experience in creating HTML documents. But the WML sides were absolutely new for us. So we first had to look for the syntax and we found out that it is very similar to the HTML Syntax. The next problem was to get the WML site running, because the server of the university does not support WML (and we did not know that).
For the implementation we took our designed user interfaces and transormed them to real user interfaces. The WML user interface supports only searching for real estates because the input and display of the other functions would not be possible on the little screen. We also included some graphics in our WML login page. Thanks to Gunnar.
The interfaces can be viewed at:

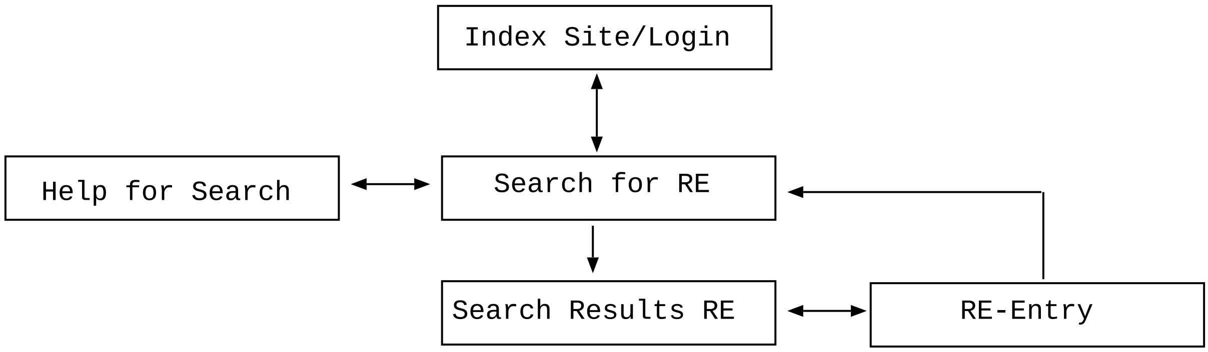
- HTML: http://www.cosy.sbg.ac.at/˜mpober/se1/se_html/index1.html
- WML: http://immowarrior.wap3.de

These are the flow diagrams of the user interfaces:

User-Interface Prototype - HTML Version



User-Interface Prototype - WML Version



6.3 The 5^{th} lesson

While we were presenting our user interface some weaknesses tourned out:

- Our user interface was optimized for high resolutions (1024x768 or more), but as we have seen this resolution is not supported everywhere and so the user interface was uncomfortably to use.
- The support requirements of our privilege system could explode if some new permissions are added.
- When searching via WAP there is no possibility to view an already entered search criteria.

We solved all problems and i want to give an explanation on the solution of the basic weakness, the privilege system. We earlier had all possible combinations of permissions listed (in a pull down menu). This was a good looking solution for our problem to this time but what happens if the nuber of different permissions has to be enlarged to 10? We will fill screens with different choices. Lucky administrator, who has the pleasure to find out the right on. So we changed it into checkboxes, having 10 checkboxes for 10 permissions.

Chapter 7

Activity Diagrams

7.1 What is an Activity Diagram?

Activity diagrams describe the workflow behavior of a system. The diagram illustrates the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. An activity is trigged by one or more events and activity may result in one or more events that may trigger other activity or processes.
Typically an activity diagram is attached to the implementation of an use case. The purpose of this diagram is to focus on flows driven by internal processing (as opposed to external events)

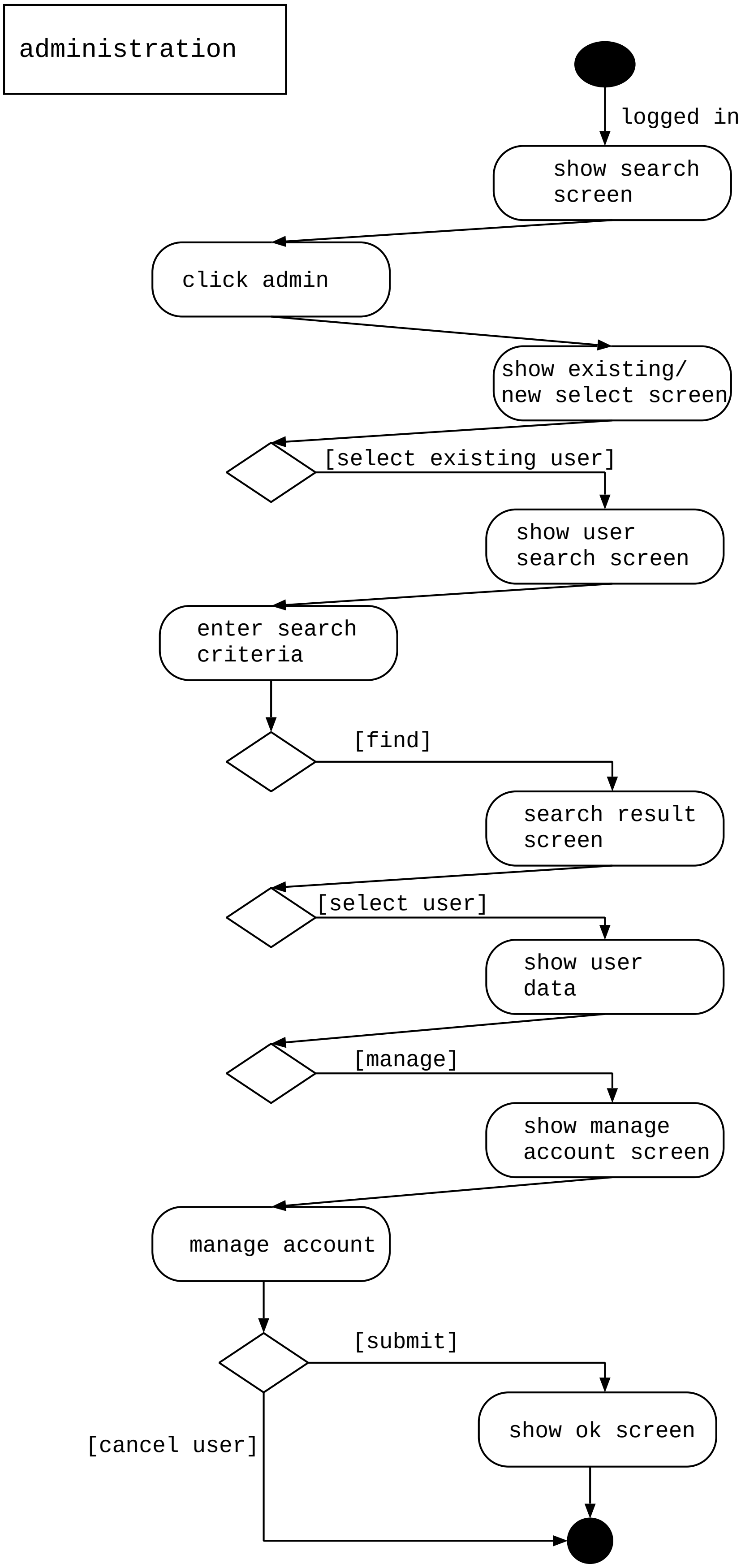
7.2 Our Activity Diagrams

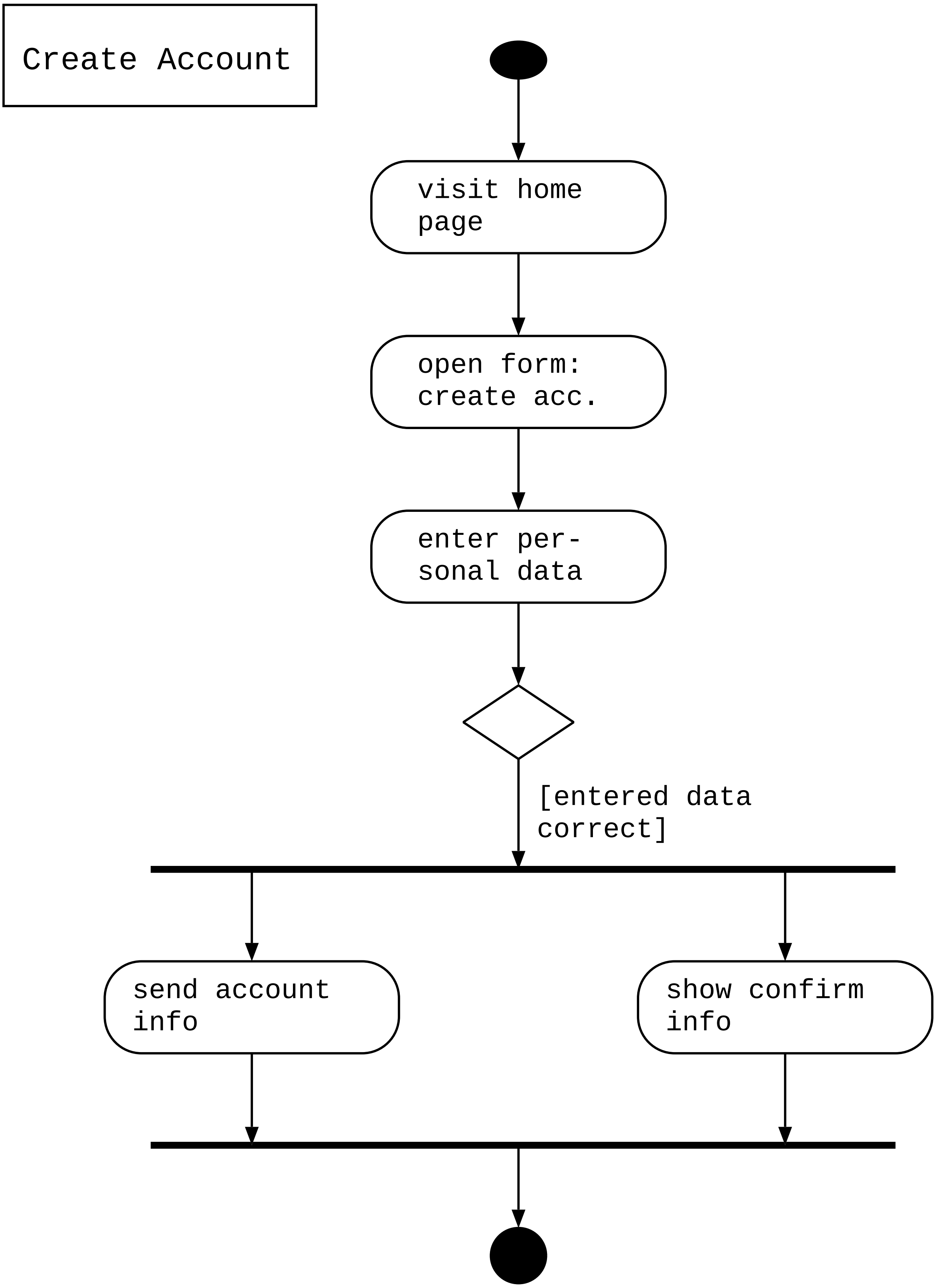
7.2.1 Our Job

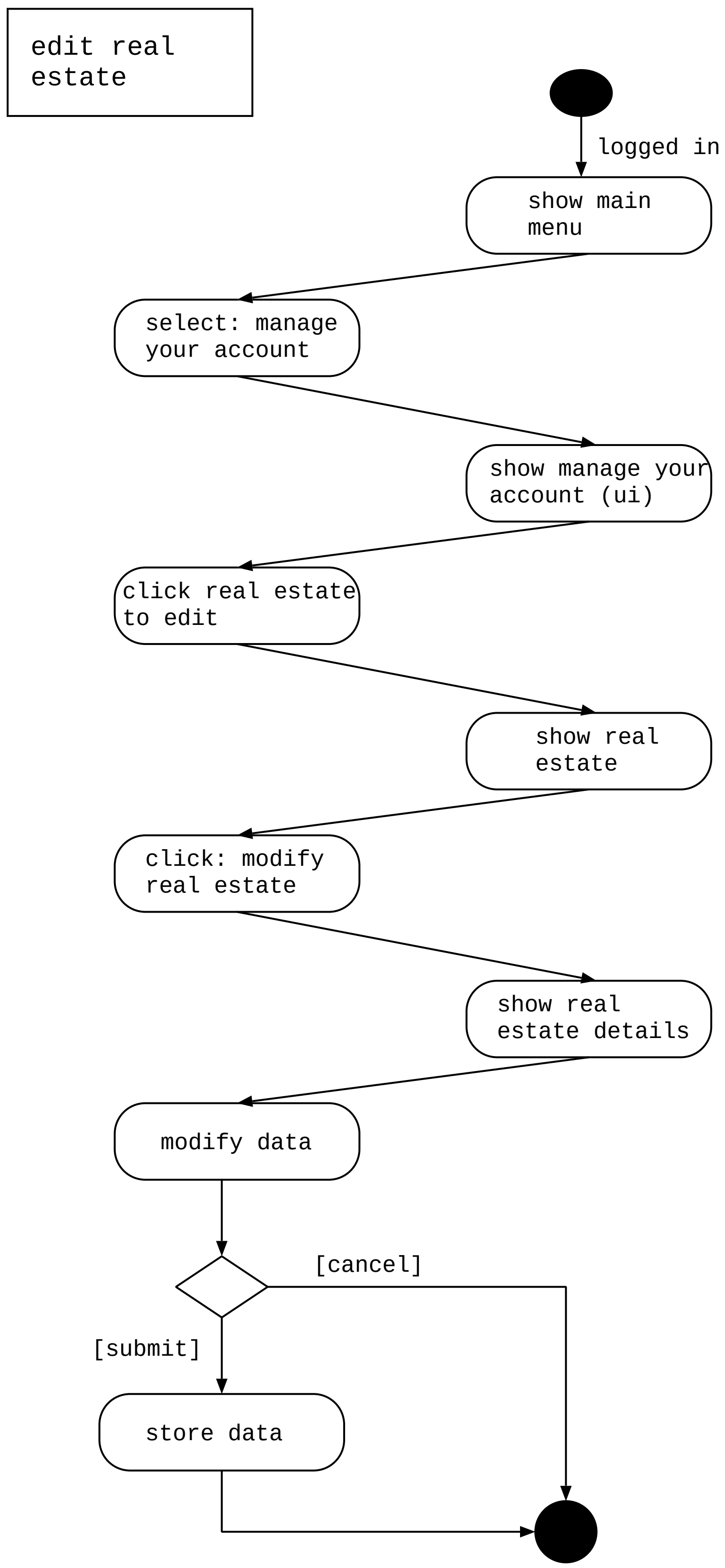
Our job was to create activity and sequence diagrams for one meeting. The meeting is described in the next chapter.

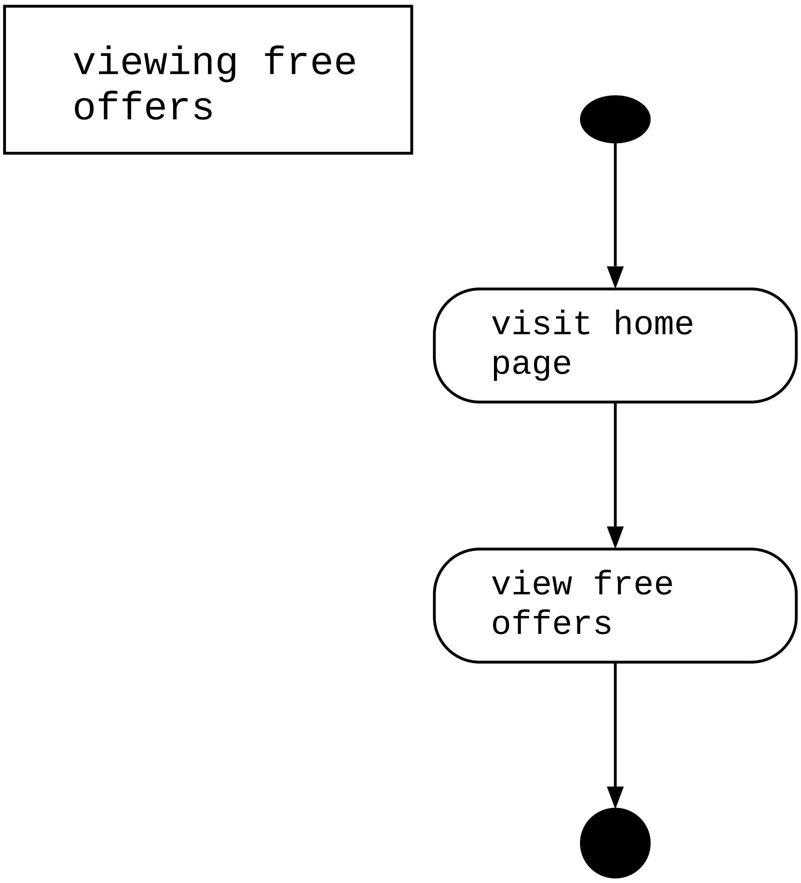
7.2.2 preperation

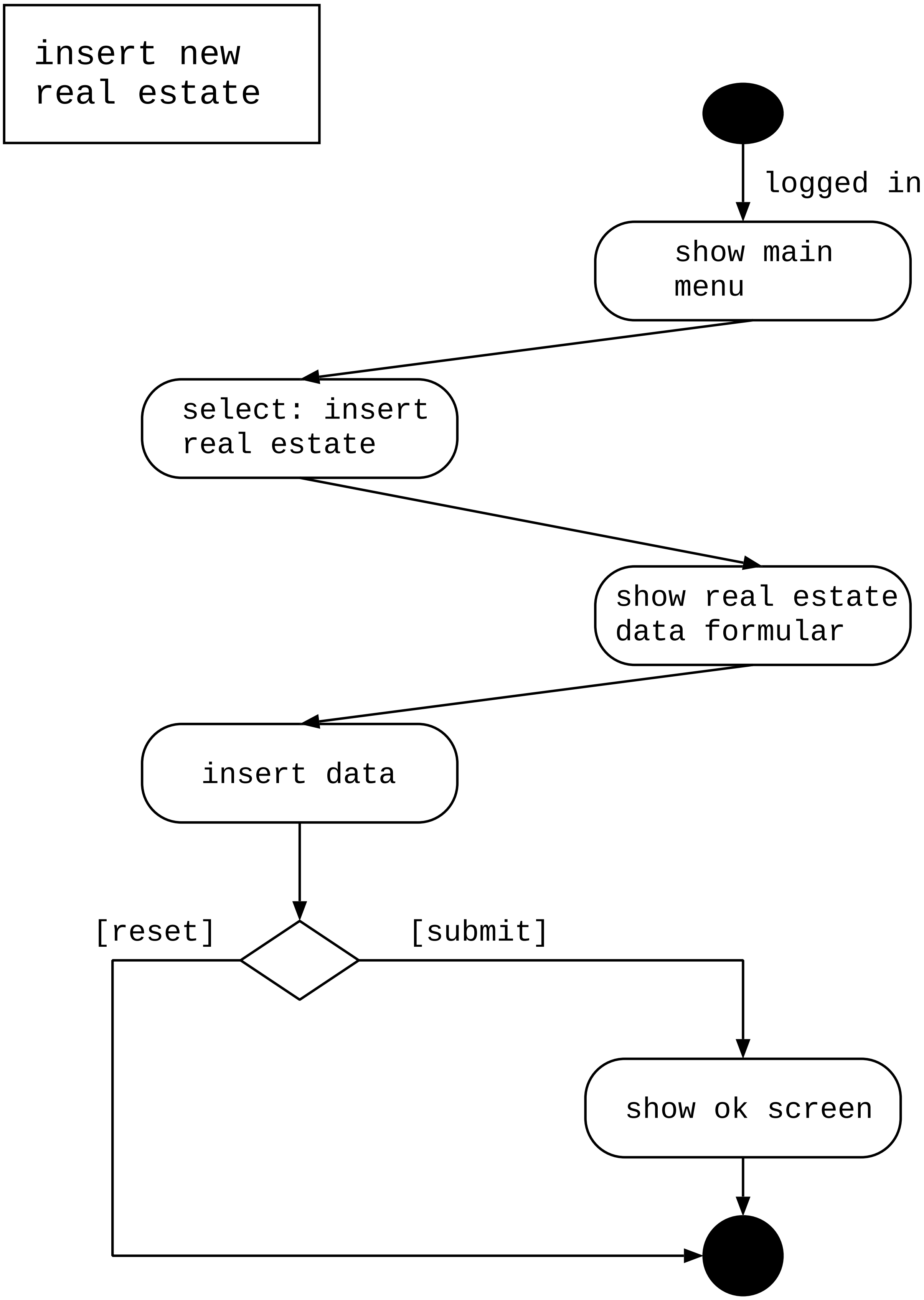
We created activity diagrams for all use cases we planned:

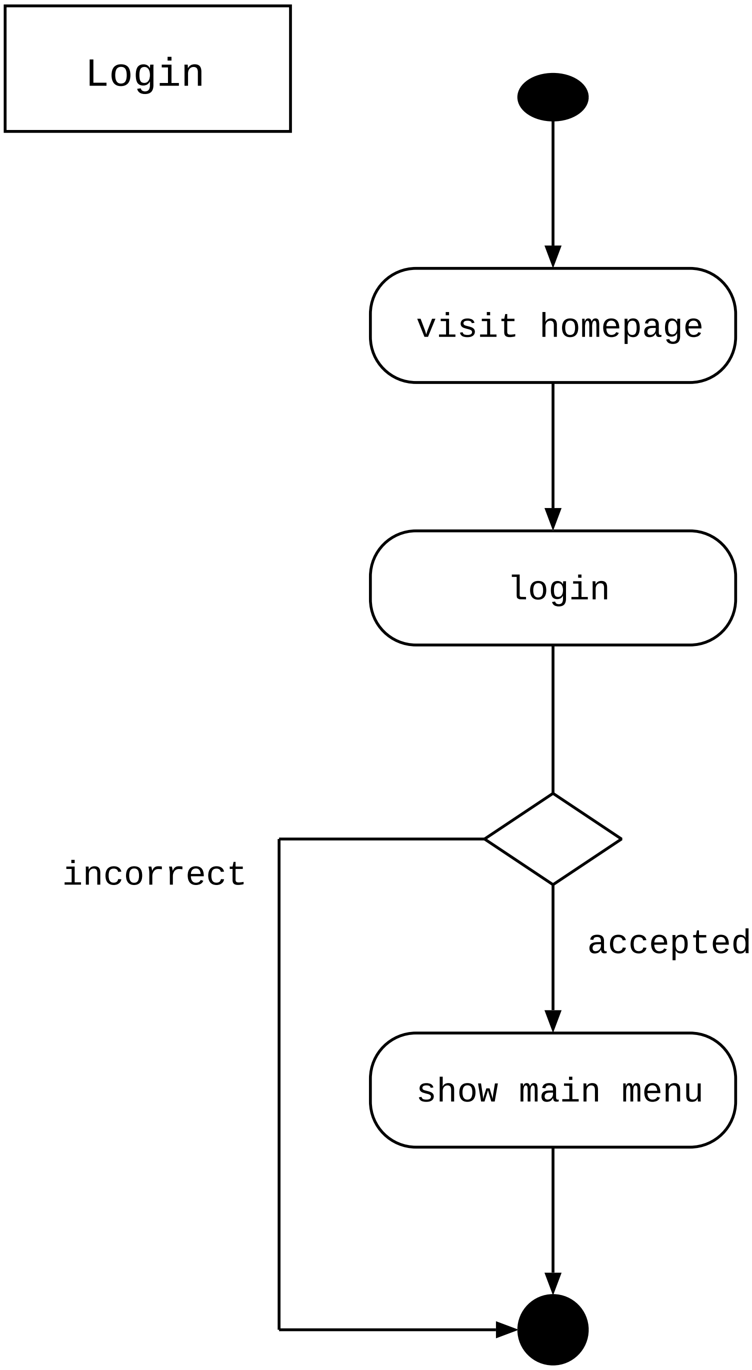

administration
logged in
show search screen
click admin
show existing/ new select screen
[select existing user]
show user search screen
enter search criteria
[find]
search result screen
[select user]
show user data
[manage]
show manage account screen
manage account
[submit]
show ok screen
[cancel user]

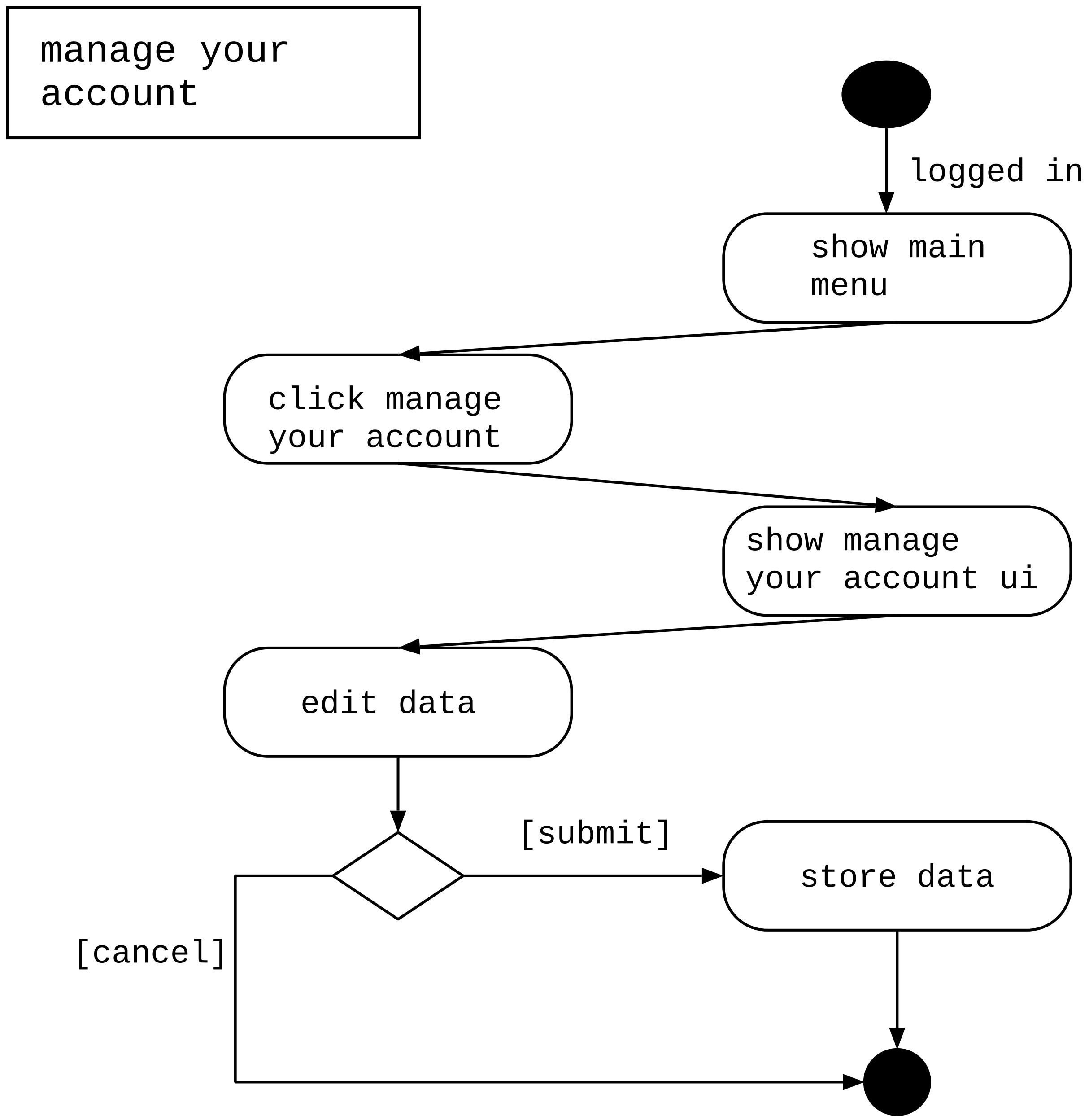

Create Account
visit home page
open form: create acc.
enter per- sonal data
[entered data correct]
send account info
show confirm info

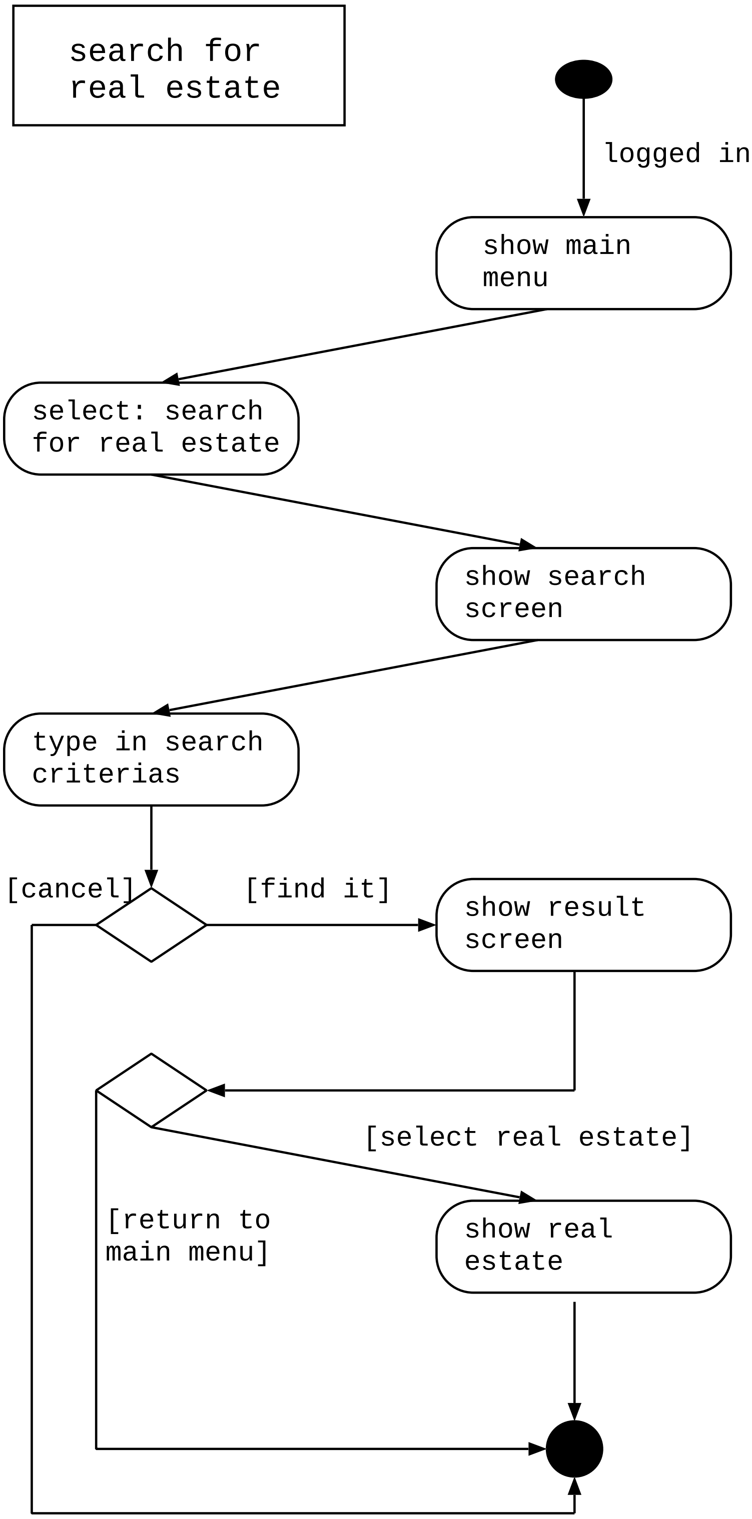

edit real estate
logged in
show main menu
select: manage your account
show manage your account (ui)
click real estate to edit
show real estate
click: modify real estate
show real estate details
modify data
[cancel]
[submit]
store data

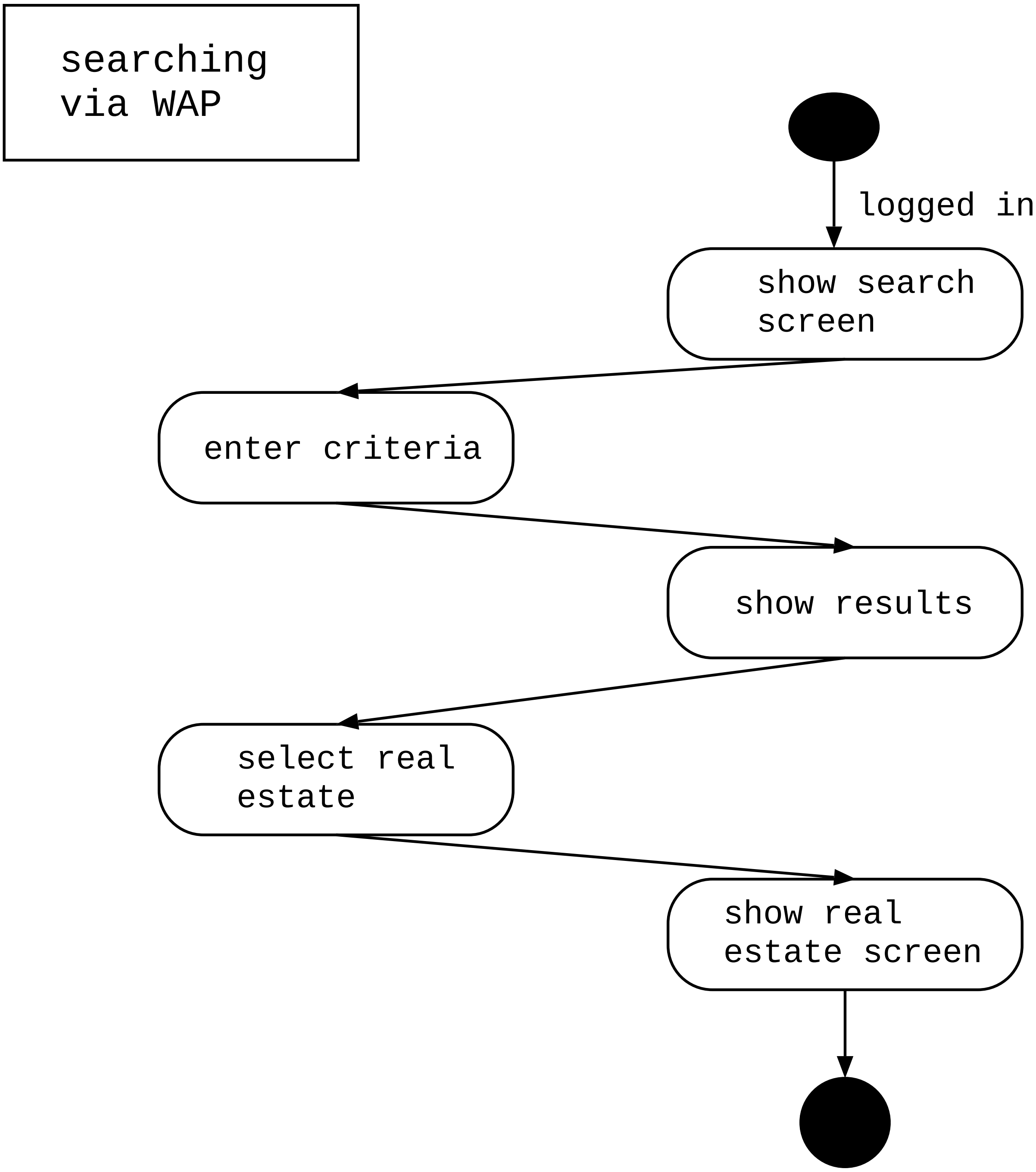

viewing free offers
visit home page
view free offers


insert new
real estate
logged in
show main
menu
select: insert
real estate
show real estate
data formular
insert data
[reset]
[submit]
show ok screen


Login
visit homepage
login
incorrect
accepted
show main menu


manage your account
logged in
show main menu
click manage your account
show manage your account ui
edit data
[submit]
store data
[cancel]


search for
real estate
logged in
show main
menu
select: search
for real estate
show search
screen
type in search
criterias
[cancel]
[find it]
show result
screen
[select real estate]
[return to
main menu]
show real
estate


searching
via WAP
logged in
show search
screen
enter criteria
show results
select real
estate
show real
estate screen

Chapter 8

Sequence Diagrams

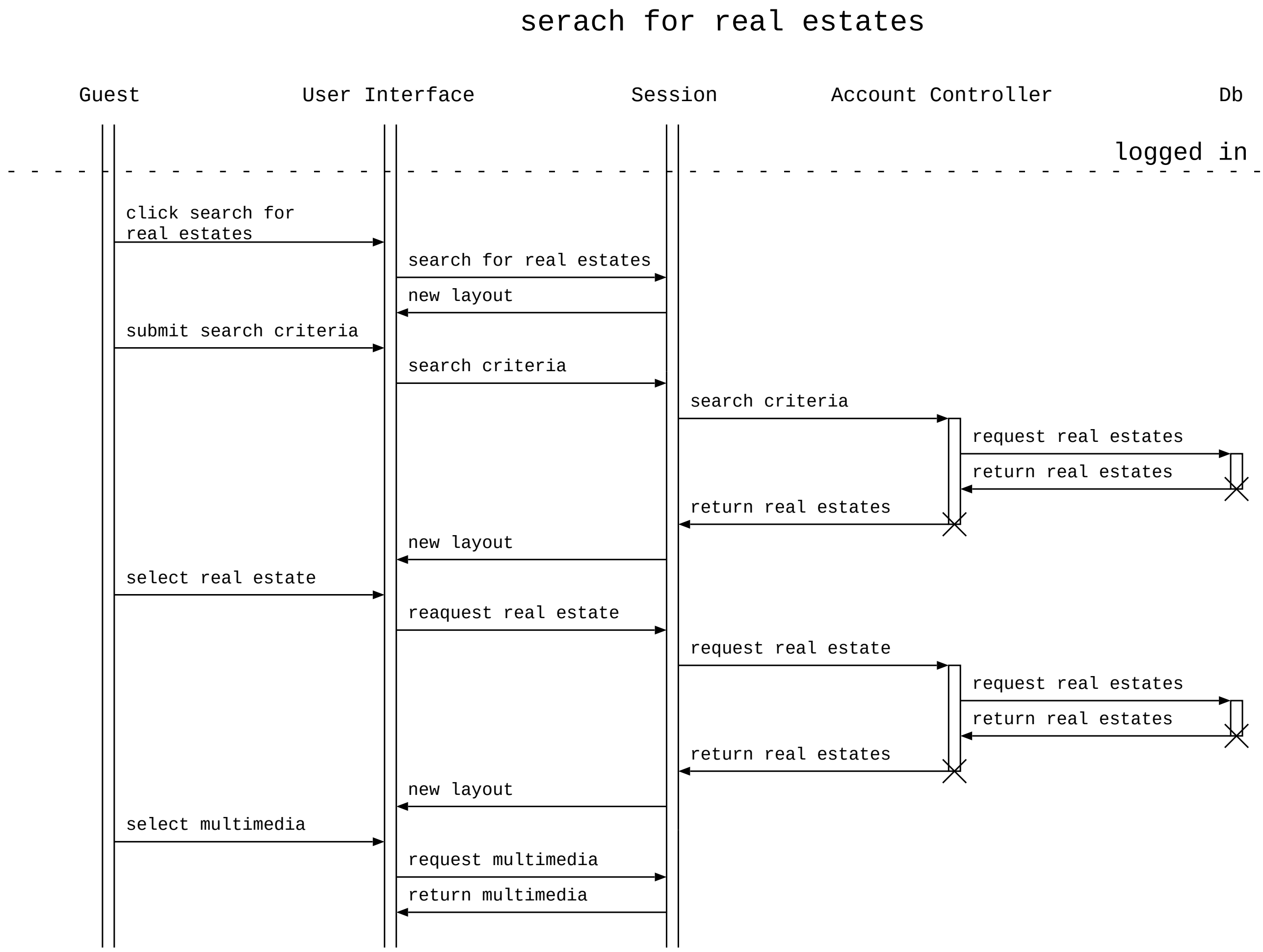
8.1 What is a Sequence Diagram?

A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their "lifelines". It does not show the associations among the objects. The sequence diagram represents an Interaction, which is a set of messages exchanged among objects within a collaboration to effect a desired operation or result.
A sequence diagram has two dimensions: the vertical dimension represents time, the horizontal dimension represents different objects. Normally time proceeds down the page. Usually only time sequences are important but in real-time applications, the time axis could be an actual metric. There is no significance to the horizontal ordering of the objects.

8.2 Our Sequence Diagrams

8.2.1 Preparation

We also did sequence diagrams for all use cases. But the sequence is alsways the same, the information is offered by the user. Then gathered from the user interface it is passed on to the session and from there stored in the database over a controller (account or real estate). So i included just one sequence diagram:



8.2.2 The 6^{th} lesson

In this lesson we presented our activity and sequence diagrams, but again they were to much to be all presented. So we had not much time to talk about improvments. The necessary improvments which came out were changes in the diagrams (not changing our system development):

- The activity diagram should be created from the view of the actor only, not from the system.
- The description of the sequence diagramm has to be written on the left side of the diagram.