
CODE REVIEW
by Team 7

for

ESE 2014 Project by Team 2

Version 1.0

S. Berger, J. Niklaus, J. Schaerer, A. Sellathurai

Contents

1	General Review	4
1.1	Design	4
1.1.1	Violation of MVC pattern	4
1.1.2	Usage of helper objects between view and model	4
1.1.3	Rich OO domain model	4
1.1.4	Clear responsibilities	4
1.1.5	Sound invariants	4
1.1.6	Overall code organization and reuse, e.g. views	4
1.2	Coding Style	4
1.3	Documentation	5
1.3.1	Understandable	5
1.3.2	Intention-revealing	5
1.3.3	Describe Responsibilities	5
1.3.4	Consistent Domain Vocabulary	5
1.4	Tests	5
2	Code Analysis: xyController	6

1 General Review

1.1 Design

1.1.1 Violation of MVC pattern

Method enlistad in AdController is huge. Sure more could be extracted that does not belong to the controller. Method getUserImage in AdController deals with images as well as enlistad. Maybe a PictureService is missing. PICTURE_LOCATION and servlet-Context belong to the PictureService as well. The same applies for enlistroomie in RoomieController. Why does does not the Adservice return directly the interessents, but they are constructed in besichtigungsterminSetzen in AppointmentController?

1.1.2 Usage of helper objects between view and model

Method enlistad in AdController uses helper object PictureManager.

1.1.3 Rich OO domain model

What is meant with that?

1.1.4 Clear responsibilities

All success messages collected in success.jsp: They don't have anything in common. Why not in individual pages? Redirects necessary but why? (user cannot interact with website during redirect.) Good use of ErrorSaver object.

1.1.5 Sound invariants

I did not find invariants in the code.

1.1.6 Overall code organization and reuse, e.g. views

exceptions folder and thus InvalidUserExceptions exists twice. But otherwise reuse and organization fine.

1.2 Coding Style

[TBA]

1.3 Documentation

1.3.1 Understandable

In general, the documentation is very understandable. It creates a clear idea of what the code does but is still not too long. Still there are some minor issues to be noticed. An example is `AdController.showAdId`: The comment isn't very understandable. It talks about "needed criteria" but it is not clear what these are. Similarly, the doc comment for `FilterAdsController.filterAdsIndex` talks about a "smaller filter" but it is not obvious what this is and why it is needed.

1.3.2 Intention-revealing

Throughout the documentation, the intention of each method is clearly shown. The comments give a brief but revealing overview about what the methods are used for, which is going to be a big help for everyone to maintain the code in the future.

1.3.3 Describe Responsibilities

Many parts of the documentation lack a clear description of responsibilities. Often the just say "...is triggered, when...". Sometimes the comments even state which element on which .jsp triggers the method. While this is really helpful when maintaining the code, it is not really the idea of responsibility driven design. If there is going to be another view to use the same method, the documentation will no longer be fully correct. An even worse case could be when a part of the system is changed and the method is not even invoked by the stated view any more. Nevertheless, most responsibilities are still clear, since they're implied by the method name and the commentary.

1.3.4 Consistent Domain Vocabulary

Most of the used vocabulary seems to be consistent throughout the documentation. This is, with one exception: the term for a person to look for an apartment. It is referred to as "mate", "yourself", "interessent", "interested user", "applicant", "person" and "roomie". There should be one single, exactly defined word for that. Apart from that, the vocabulary is very accurate and adds to quite understandable documentation.

1.4 Tests

[TBA]

2 Code Analysis: xyController