Advanced Software Design
Project Guidelines

First iteration: 2015, Dave Clarke
Update 2016-2017: Kiko Fernandez-Reyes

October 19, 2017

This document gives guidelines for the Advanced Software Design project. The project involves the (partial) design and implementation of the MATCH software infrastructure. The requirements for this software are described in the document “Advanced Software Design: The MATCH System Specification.” Assessment will be based on a newly developed scheme called achievement-driven learning. This scheme is described in more detail in the document “Advanced Software Design: Achievements.” In that document there is a list of achievements that need to be demonstrated to a teaching assistant in order to obtain each particular grade for the course.

1 Guidelines

Here are a number of guidelines to follow when developing your design.

1.1 General Guidelines

1. The system-under-design is deliberately very large and you are not expected to produce a complete design for all of it, especially if you wish to make it flexible enough to handle all possible configurations. This means that certain choices will need to be made in your design process to limit what aspects of the system are considered in detail. Your design should support most of the MATCH functionality to some degree and support one MATCH configuration in sufficient detail.

2. A good starting point is to flesh out a domain model to better understand the concepts in the system. If you are more implementation oriented, an alternative strategy is to implement a skeleton of the system to get a feel for the concepts involved.

3. To be faithful to the MATCH requirements, your system needs to consist of a server, which is where most of the complication resides, and two clients (a responder and a requester).

4. A good strategy is to first design a version of the system that supports end-to-end functionality for a single configuration, where most features are mocked up. Gradually this basic design can be improved by replacing mock designs by more realistic ones. Then, to support further configurations, features can be generalised.
5. Select a subset of the possible features that you want to design and develop the smallest thing that could possibly work to provide those features. Expand the system incrementally, always considering the design. Identify design flaws (document them) and try to fix them.

6. The system-under-development can vary in two dimensions: features and flexibility. Many features can be considered optional or can have rudimentary designs for the early versions of the system. Many features can be designed to varying degrees of flexibility: a feature can have a mock or trivial design, a design that supports one possible variation, or a design that supports all (or most) variations.

Your design will need to support a reasonable number of features to design to pass. Higher grades require more flexible designs of some features.

7. Ultimately, your design must be sufficient to support the process of demonstrating achievements—almost all achievements relate to the system-under-design.

8. During the course you will learn ways to evaluate and improve your design. These will enable you to improve your design as the course progresses. It is important to keep track of what changes you make and why.

1.2 Implementation Guidelines

1. Technically, no complete implementation is required, though some achievements require some code to demonstrate that you understand the connection between design models and code.

2. Nevertheless, it is a good idea to implement a prototype of your system under design. Your implementation does not need to be industrial strength. Implementing the system is a way to test out your designs and to identify design flaws. It is advised to implement incrementally and to interleave design and implementation activities. In any case, we are more interested in the design than the implementation.

3. Whenever you feel stuck, implement something. An implementation can help guide the design and can help resolve design challenges.

4. There are no constraints on how your prototypes are implemented—they could be mobile apps, desktop apps, web apps, or even command-line apps.

1.3 Assessment

1. The design work that you do will be evaluated by the teaching assistant in the weekly meetings. The teaching assistant will base his opinion not only on the actual design you produced but also on your justifications for why your design is the way that it is. Documenting these justifications will thus greatly improve your interactions with the teaching assistants and your final results.

2. The teaching assistants keep track of the whether everyone in the team is participating during meetings. Individuals failing to participate will miss out on the achievements.

3. Expect to not pass the achievements the first time you attempt them. This may be frustrating, but it’s perfectly reasonable. At the start of the course you do not know very much about good design—you learn this knowledge as the course progresses.
4. It costs nothing to try an achievement and fail. In the process you gain feedback.

2 Contact Moments

There are two kinds of contact moments related to the assignment: weekly meetings with TAs and the final design review.

Weekly Meetings  Each week there will be a meeting with a teaching assistant (TA) to obtain assistance, discuss your design, to obtain feedback, and to demonstrate achievements. Meetings will last for 30 minutes. Weekly meetings are the time to get additional help.

A web software system has been developed to track your progress through the achievements. The following guidelines will help you use your weekly meetings wisely.

- 24 hours before the meeting, you need 1) inform the TA which achievements you wish to demonstrate, and 2) send the TA your current design documents.

  The web software can be used for this purpose. Please, respect the 24 hours deadline or you will not be able to tick off achievements that week.

- You need to come up with a plan of how to demonstrate the achievements. This does not need to be written down, but it must be clear to the team — please don’t bring a script to read. Demonstrations must be concise, so you need to know what you are talking about.

- Achievements can only be demonstrated at the start of the meeting. The remainder of the meeting can be used to improve your design and understanding.

- You are entitled to one 30 minutes meeting per week. Use the time wisely.

  Arrange the meeting times personally with your TA based on your availability and the TA availability times. The time can change weekly if required!

Final Design Review  The goal of the final design review is to review the design of another team and compare it to your team’s design. Show us that you have a deep understanding of the problem, that you can understand others’ designs, and that you can question them appropriately. We are more interested in semantic errors than notational bugs! Similarly, don’t quibble over choice of words, unless it actually makes a difference to the interpretation.

  You are expected to hand in your design prior to such sessions and to study the other group’s design. You will be provided with another group’s assignment to review.

3 Deliverables

You need to submit your ongoing design documents to the teaching assistants each week. The designs submitted are not graded, but are used for obtaining feedback and demonstrating achievements. Most achievements cannot be demonstrated without supporting design documentation.

In order to participate in the final design review, you need to submit your design documents (to your TA) by the following deadline.

  Due date final design: Midday, Friday 8th December, 2017.
The final version of the design will be used by another team in order to perform a comparative analysis of the designs. This document needs to be much better presented. It is a good idea to present your designs as a textual document describing the diagrams and other artefacts (which may appear as appendices referred to by the text.) The text itself need not be overly long, though it should highlight key aspects of your designs and indicate key choices made, to make it understandable to a third party.

The design review will take place during week 50 (from 11th until 15th of December, 2017.

If it is judged by the lecturer and teaching assistants that your document is inadequate for others to review, due, for example, to severe incompleteness or poor presentation, then your team will not participate in the review process, which will affect your ability to demonstrate all achievements. This applies also if your design is delivered late (without prior negotiation).

Failure to submit design documents will make it difficult to demonstrate certain achievements. Thus submitting something is always better than submitting nothing.

4 Modalities

1. You will work on the assignment in a group with four members. If the size of your team drops below 3 members, your team will be merged with another team.

2. You are strongly encouraged to create groups with members of mixed backgrounds (programmes, experience, Swedishness, ...).

3. If you have not signed up for a group before the second lecture, you will be assigned to one — consult one of the TAs, Kiko (kiko.fernandez@it.uu.se), Albert (albert.yang@it.uu.se) or Nam (hainam2511@gmail.com).

4. Your time should not be spent getting as much of the system design completed as possible. The design activity must be seen as a vehicle for learning about software design, and as a means by which the group will be able to demonstrate achievements.

5. Deadlines are absolute, unless negotiated well in advance (more than a week beforehand).

6. In addition to the final design document, you are required to provide all of your auxiliary documents and code along with your submission for the team reviewing your design to examine.

Final Deadline

The last date to tick off achievements is 18th December, 2017
Do not leave everything until the last minute.