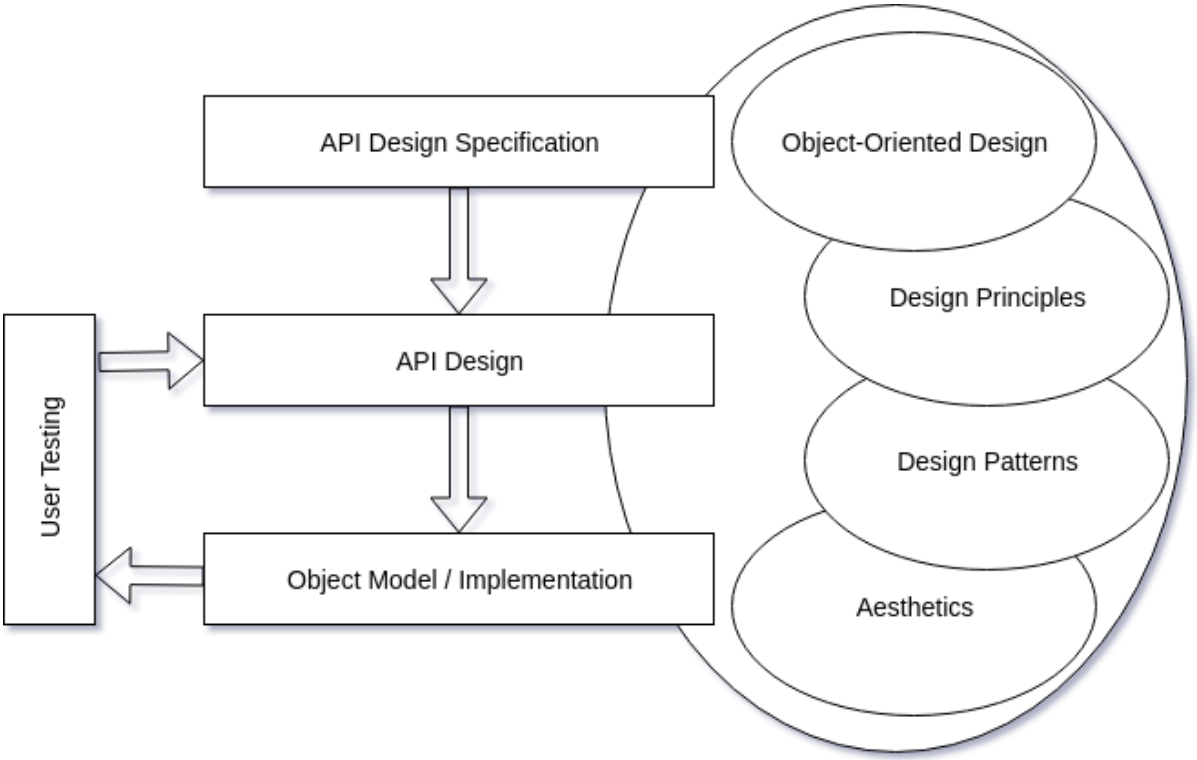
Sensorveiledning

Rammeverk ITF20119

Om Kurset

Fokuset i kurset har vært på forstå rammeverk og deres oppbygning ved å utvikle et eget rammeverk ved hjelp av gitt teori for kodedesign og en strukturert designprosess. Designprosessen er kalt scenario-driven design og er illustrert under. Firkantene symboliserer konkrete steg i prosessen og samsvarer med innleveringer som er blitt gjort i gjennom semesteret. Ellipsene symboliserer teoretiske emner som er blitt forelest gjennom kurset.



En komplett liste over forelest teori og oppgaver finnes på Canvas-sidene til kurset.

Om Eksamen

Eksamen består av en prosjektinnlevering med kode og rapport.

Studenter har jobbet individuelt eller i grupper, men det skal leveres individuelt i Canvas. Studenter som har jobbet i en gruppe kan levere samme kode og rapport, eller de kan velge å skrive individuelle rapporter som beskriver sitt eget bidrag.

Under er en kopi av oppgaveteksten for sluttleveringen på Canvas.

The final project will count for the entire grade in the course.

The project can be a group effort, but each student must deliver individually.

On delivery you must provide the code and documentation written for your framework and a report documenting the design (and functionality).

The Report
The report count for 60% of the final grade. It can be the same for members of a group if the workload has been roughly similar, but each student can also write their own report to highlight their own contribution. Shared sections are allowed between the individual reports to describe the overall project.

The report should be a pdf-file between 10 and 30 pages long (and estimate) and contain a similar structure to the following.

Introduction
A high level description of the framework that you are creating (can be shared in group projects). Focus on the functionality that you want the framework to provide.
A brief description of the project group (can be shared)
Background
A background section to establish a record of existing solutions and what they can and cannot do, and how the created framework fits into this picture (can be shared). Client-code for something similar to your own scenarios might be a good idea to highlight certain areas.
Method
A section to describe the method under which the framework has been created (can be shared). This section should contain a description for each of the stages given in the "Framework Design" lecture (API Design Specification, API Design, Implementation and User Testing).
You should describe how you have worked, not what you produced, in this section.
The Design Process/Results
This section should give a description of how the API has changed throughout the process. This has likely been a messy process, and you've might have started completely off base. For the sake of this report you should clean it up to give a better picture.
Important stages are
API Design Specification: Dream/pseudo code and a short discussion/reasoning for each of the scenarios
API Design. Describe the interface created based on the specification and what principles/patterns were taken into account
Implementation. Describe the implementation created for the established interface. Describe/discuss the design principles/patterns that have been involved.
User Testing. The test results can be structured around the scenarios and should provide a description/discussion of the feedback and changes.
For group projects each student must write about the scenarios for which they have contributed. If some are shared between members the workload has to be documented.
Resulting Framework
A section to present the resulting framework. Present the final code for the selected scenarios and provide a high-level description of the overall solution. The section should include a conclusion of how design patterns and principles and user feedback have been incorporated into the solution
Discussion
A section to discuss the project and any conclusions (can be shared)

What went well? what could have worked better?
The Framework
The framework delivered should look like a working third-party framework that can be used by any developer.

The code, code should be identical for all group members,
Documentation
Getting Started
The implementation should ideally contain tests, but it is not a primary evaluation criteria when setting the grades.

Large compiled files should not be a part of the delivery.

Vurdering

Overordnet

- Er det utviklede prosjektet et programmatisk rammeverk framfor en applikasjon?
- Har rammeverket blitt utviklet ved å følge en strukturert design prosess (scenario driven design og user testing)?
- Er hensikten med, og abstraksjonsnivået til, rammeverket definert tydelig?
- Er det leverte materialet (rapport og kode) ryddig og skikkelig presentert?

Rapport (60%)

I tillegg til de spesifiserte kravene til struktur og innhold gitt over skal følgende vurderinger gjøres.

- Har rapporten riktig perspektiv og fokus?
    - Skal gi en god oversikt over hele utviklingsprosessen slik som skissert i figuren i begynnelsen av veilederen
    - Rapporten er IKKE en arbeidslogg men akademisk tekst som skal presentere det relevante innholdet på en lettlest og oversiktlig måte
- Gir rapporten et god oversikt over hvordan rammeverket har utviklet seg gjennom designprosessen basert på forelest eller relatert teori for kodedesign?
- Gir rapporten en god oversikt over det endelige rammeverket?

Rammeverk / Bibliotek / API (40%)

Det viktigste aspektet av koden er APIet. Det er påkrevd at deler av koden skal fungere, men ikke at hele løsningen fungerer
I tilegg til selve koden så skal det leveres en getting started guide og dokumentasjon.
Alle design-prinsipper og -mønstre forelest er relevant for vurderingen av koden, men her er noen fundamentale vurderinger.

- Lekker implementasjonsdetaljer ut i grensesnittet?
- Er navnene lettleste? Uforståelige forkortelser?

- Er navngivningen av klasser, metoder, etc. konsekvent?
- Følger grensesnittet konvensjonene til språket rundt?
- Er grensesnittet konsistent?
- Er parametrene konsistente mellom metodene, spesielt de som gjør lignende ting?
- Gir metodene mening uten eksplisitt kall til setup/teardown?
- Er det lange parameterlister til enkelte metoder/konstruktører som burde ha vært konfigurasjonsobjekter eller builder-patterns i stedet for å tydeliggjøre funksjonaliteten?
- Gir parameter- og returtyper mening, og representerer de den faktiske datatypen som gis tilbake?
- Er det eksplisitt designet for hva som skal tillates av arv av rammeverksklasser, hvis mulig?
- Er tilgangene i rammeverket minimalisert - mao. rr ting private/protected/public med tanke på hva som skal være tilgjengelig og bruksområde for brukeren?

Karakterer

For å få en C karakter må studenten:

- ha levert et prosjekt som kan ansees som et rammeverk/bibliotek basert på etablert teori for kodedesign og språkkonvensjoner
- ha gjennomført en designprosess med ikke mindre enn 5 ikke-trivielle* scenarier
- skrevet en rapport som gir en god oversikt over både prosess og resultat
- deler av koden fungerer og APIet ser fornuftig ut
- Har skrevet en grei dokumentasjon og en grei getting started guide

For å få en E karakter må studenten:

- ha levert et prosjekt som kan ansees som et rammeverk/bibliotek basert delvis på etablert teori for kodedesign og språkkonvensjoner
- ha gjennomført en designprosess med flere enn 2 ikke-trivielt* scenario
- skrevet en rapport som gir en oversikt over både prosess og resultat og oppfyller de spesifiserte kravene
- Lite men noe fungerende funksjonalitet og et API som ligner på noe fornuftig
- Har en dokumentasjon og getting started guide

Et ikke-trivielt scenario er komplisert nok til å tillate en viss kompleksitet i koden og sjeldent mindre en 5 linjer kode

Karakterbeskrivelser

A - Fremragende - Fremragende prestasjon som klart utmerker seg. Kandidaten viser svært god vurderingsevne og stor grad av selvstendighet.
B - Meget god - Meget god prestasjon. Kandidaten viser meget god vurderingsevne og selvstendighet.

C - God - Jevnt god prestasjon som er tilfredsstillende på de fleste områder. Kandidaten viser god vurderingsevne og selvstendighet på de viktigste områdene.
D - Nokså god - En akseptabel prestasjon med noen vesentlige mangler. Kandidaten viser en viss grad av vurderingsevne og selvstendighet.
E - Tilstrekkelig - Prestasjonen tilfredsstiller minimumskravene, men heller ikke mer. Kandidaten viser liten vurderingsevne og selvstendighet.
F - Ikke bestått - Prestasjon som ikke tilfredsstiller de faglige minimumskravene. Kandidaten viser både manglende vurderingsevne og selvstendighet.