

Modelling

Agenda

- ▶ Hva og hvorfor modellere
- ▶ Klassediagram
- ▶ Sekvensdiagram
- ▶ Tilstandsdiagram

Hva og hvorfor modellere

- ▶ Å modellere er å lage visuelle diagrammer som illustrerer hvordan
 - ▶ Deler av systemet fungerer (flyt)
 - ▶ Komponenter i systemet henger sammen (arkitektur)
- ▶ Modellering er en enkel måte å tenke på systemdesign uten å skrive kode
 - ▶ Finne/løse problemer tidligere
 - ▶ Utgangspunkt for diskusjon
 - ▶ Støtter felles forståelse

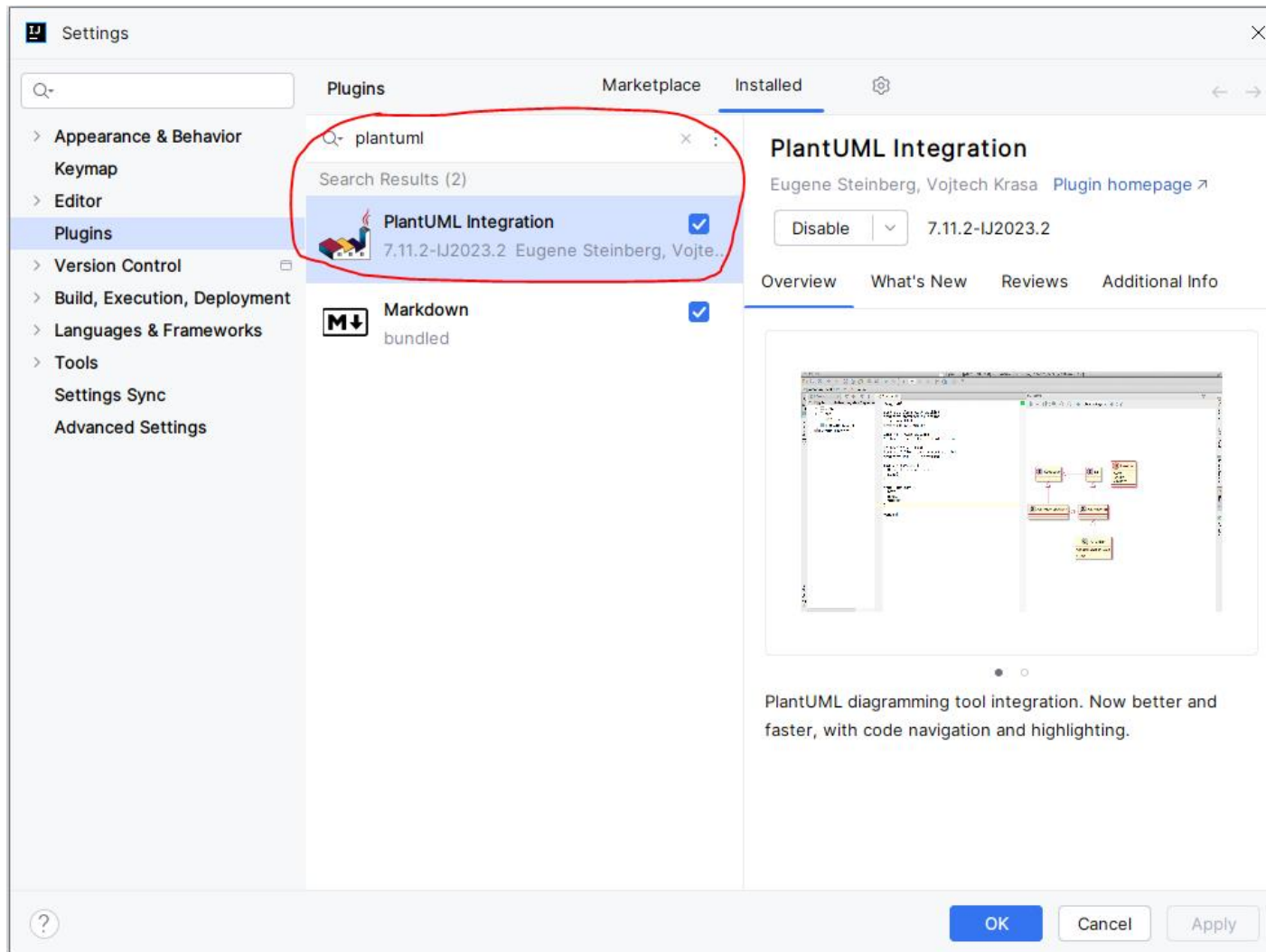
PlantUML



- ▶ Det finnes mange verktøy for å modellere (basert på UML)
 - ▶ PlantUML
 - ▶ LucidChart
 - ▶ draw.io
 - ▶ ...
- ▶ PlantUML er et kompileringsbasert alternativ
 - ▶ Definerer diagrammet som «kode»
 - ▶ Generer det visuelle diagrammet for oss
 - ▶ Finnes som plugin i IDE-er (blant annet IntelliJ IDEA)
 - ▶ Kan generere klassediagrammer fra kode
 - ▶ Gratis

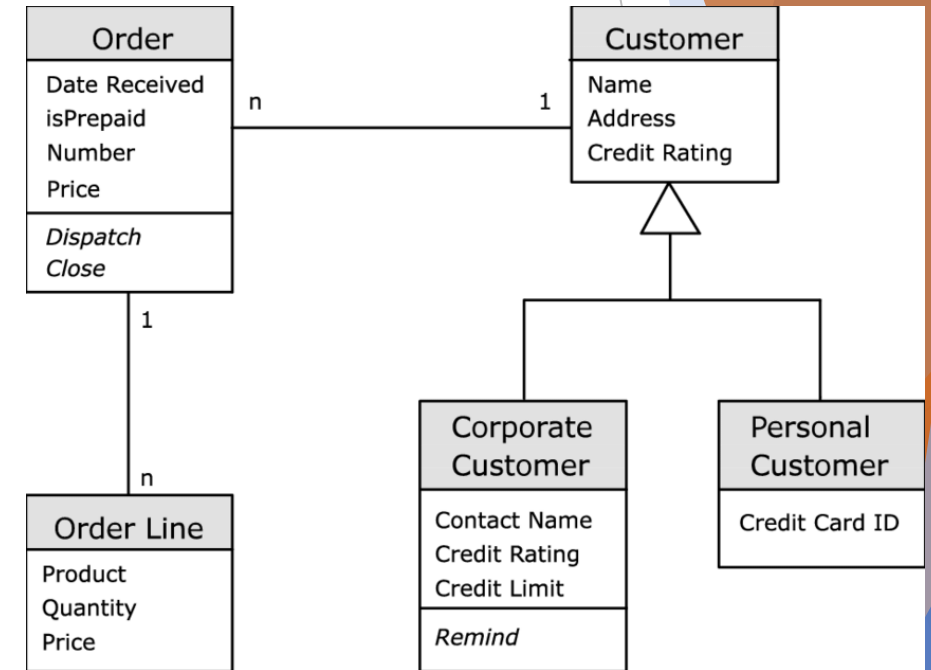
PlantUML i IntelliJ IDEA

- Krever installasjon av en PlanUML som en plugin



Klassediagrammer

- ▶ Modellerer klassene i et system og forholdet mellom dem
 - Arv
 - Implementasjon (av interface)
 - Referanse
 - Aggregering
- ▶ Vi kan komme frem til klasse-struktur før vi begynner å kode
 - Samkjører utviklere

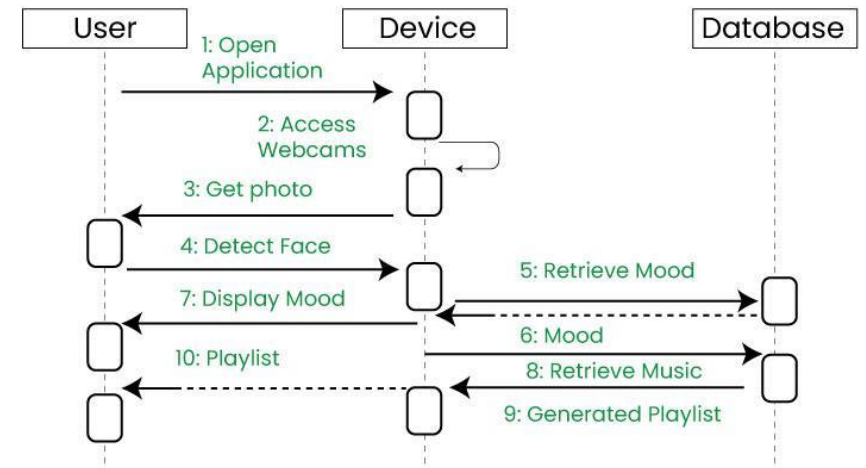


https://www.researchgate.net/figure/A-simple-class-diagram-for-a-commercial-software-application-in-UML-notation-The_fig3_225686440

Sekvensdiagrammer

- ▶ Modellerer kommunikasjonsflyt mellom forskjellige komponenter
 - De relevante komponentene
 - Rekkefølgen av kommunikasjon
 - Gir inntrykk av data som overføres
 - Kan evt. vise konkrete funksjonskall
- Hjelper med å samkjøre komponent-kommunikasjon
 - Ekstremt viktig!
 - Belyser krav til funksjonalitet i de forskjellige komponentene

Example sequence diagram



Sequence Diagrams

<https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>

Tilstandsdiagrammer

- ▶ Viser flyten mellom tilstander og hendelser i systemet
 - Kan beskrive handlingsmønstre og håndtering
- ▶ Nyttig for å drøfte eller forklare ...
 - Brukerhandlinger og valg
 - Betinget kjøring av funksjonalitet
 - Annen algoritme-lignende logikk

