



# Systeemanalyse

Project Jabberpoint

03-02-2025

Versie 1.0

—

## **Software Quality**

Bram & Kimmy  
van Schaikweg 94  
7811 KL Emmen

# Systeemanalyse

Het aangeleverde systeem bevat een aantal opvallende zaken die op het eerste oog aangepast kunnen worden.

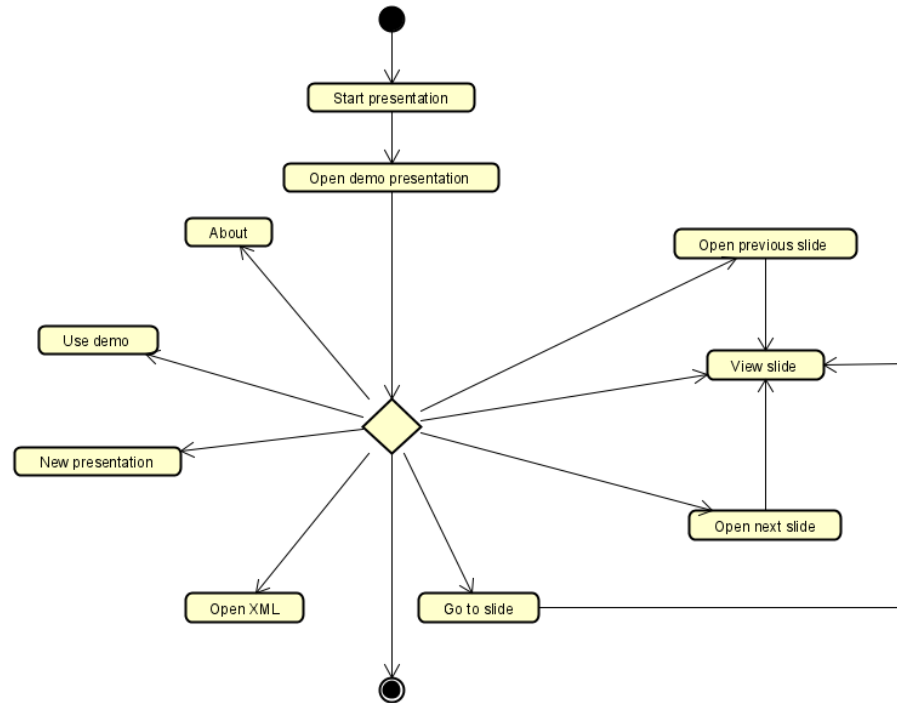
- Het systeem bevat veel onnuttige comments terwijl erbij veel andere code wel comments nodig zijn.
- Er zijn veel bestandsnamen en variabele namen onduidelijk waardoor het lezen van de code moeilijk maakt.
- De packages die er zijn, zijn ook erg onduidelijk, er wordt veel gebruik gemaakt van afkortingen terwijl dit het project onoverzichtelijk maakt.
- Veel velden of methodes hebben geen access modifiers.
- Er zijn geen testen aanwezig.
- This keyword mist vaak in een return statement.
- In de model klasse, staat een constructor tussen de fields.
- Er is code uitgecomment, dus dit kan worden verwijderd of aangepast.
- Er wordt geen gebruik gemaakt van een consistente code conventie.
- Er staat op sommige plekken een @override boven een methode terwijl die niet is geïmplementeerd vanuit een hoofdklasse.
- Op sommige plekken wordt er niet goed gebruik gemaakt van een constante.
- Veel indentaties zijn inconsistent of slecht.
- Er zijn verschillende methodes met te veel functionaliteiten, het kan ook opgedeeld worden in verschillende methodes.
- De code kan niet worden uitgevoerd.
- Op sommige plekken mist er validatie van data en ook testen die daarbij horen.
- In "MenuController", wordt er gebruik gemaakt van een god methode, en of het niet volgen van de code conventies, er worden geen enters gebruikt om code af te bakenen waardoor het heel lastig te zien is wat bij wat hoort.
- Op sommige plekken wordt er geen gebruik gemaakt van getters en setters.

Zie onderstaande diagrammen voor een overzicht van de aangeleverde code.

Afbeelding 1 is een activity diagram, met daarin verschillende keuzes die kan worden gemaakt in Jabberpoint. Daarna is afbeelding 2 een use case diagram. Vervolgens is er een klassendiagram te vinden en een sequence diagram van de main klasse.

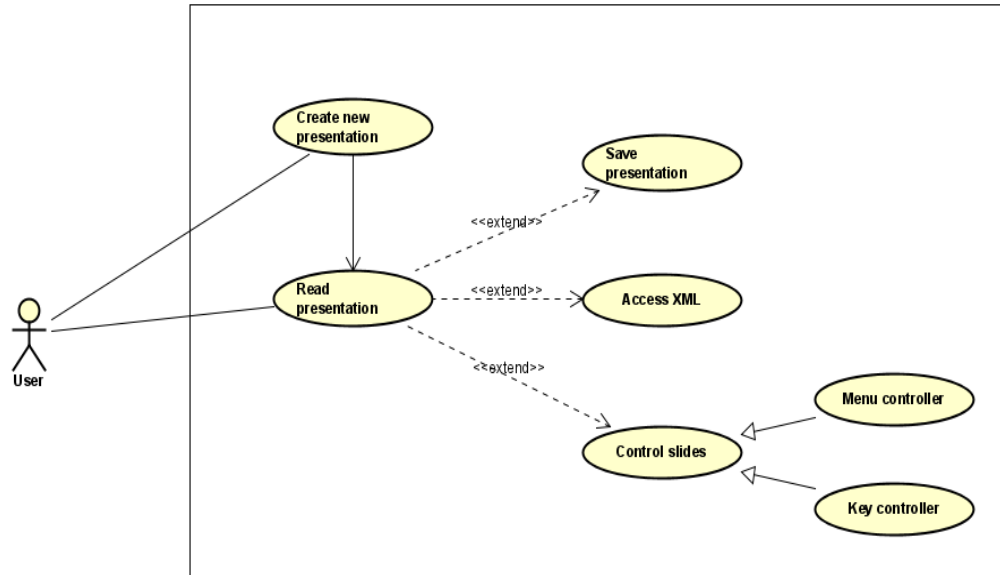
# Activity diagram

act Activity Diagram0



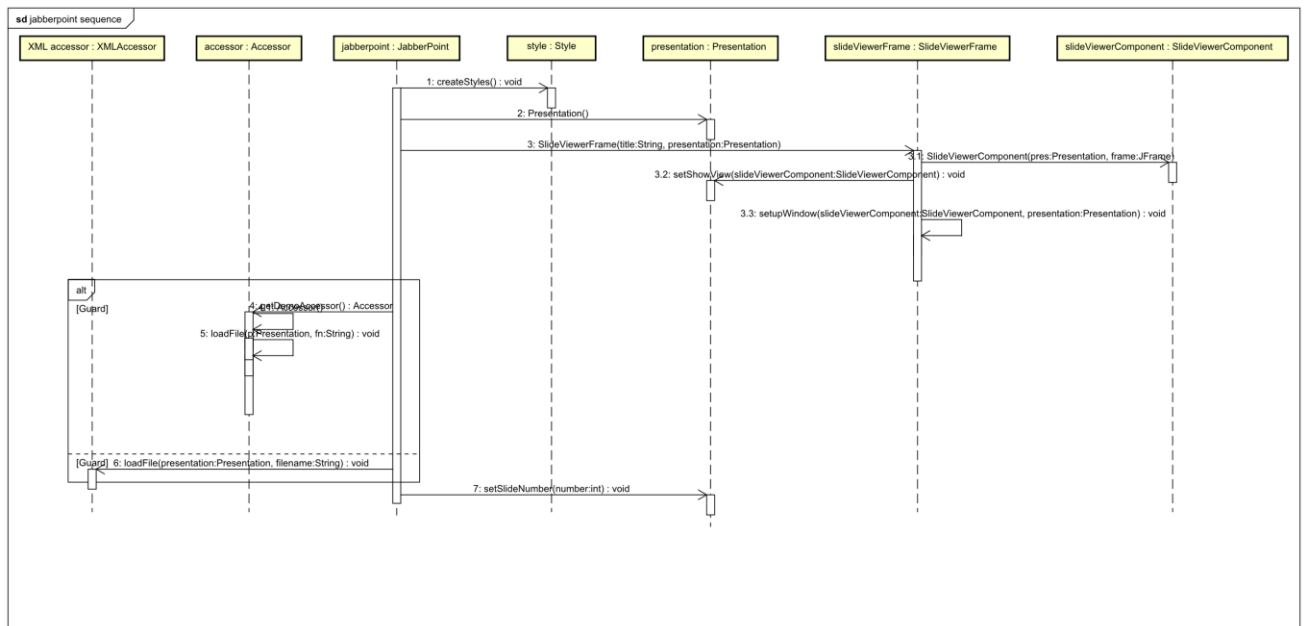
# Use case diagram

uc Use case diagram



[illegible]

# Sequence diagram



# Analyse per klas

## AboutBox

### Show methode

- Opent een dialoogvenster met informatie aan de hand van JOptionPane.

## Accessor

Een abstracte klasse met de volgende informatie

- Bevat 2 constanten voor de demo naam en de standaard extensie.
- Heeft een statische methode die een nieuwe demo presentatie teruggeeft.
- Bevat een lege constructor.
- Heeft 2 methodes om een bestand te laden en een bestand op te slaan, gebruikte beide dezelfde IOException.

## BitmapItem

Maakt gebruik van de superclass SlidelItem

- Heeft de volgende fields een BufferedImage en een imageName.
- Bevat 2 constanten voor het bestand en voor niet gevonden.
- Heeft een constructor die een bestand zoekt, als die niet gevonden is dan wordt er een error geprint.
- Bevat een lege constructor.
- Heeft een methode om de naam terug te geven.
- De methode getBoundingBox berekent de ruimte die de afbeelding nodig heeft en geeft deze terug als Rectangle.
- De methode draw geeft niks terug maar schrijft het plaatje op de gegeven Graphics.
- De methode toString toont alle fields.

## DemoPresentation

Maakt gebruik van de superclass Accessor

- Overschrijft de methode loadFile vanuit de hoofdklas Accessor. En doet het volgende:
  - o Zet de titel als Demo presentation.
  - o Maakt een nieuwe slide aan.
  - o Zet alle informatie in de slides.
  - o Voegt de slide toe aan de presentatie.
  - o Doet dit vervolgens voor nog 2 slides.
- Overschrijft ook de methode saveFile vanuit de hoofdklas Accessor. En doet het volgende:
  - o Gooit een nieuwe IllegalStateException. Waaruit blijkt dat je geen demo presentatie mag opslaan.

## JabberPoint

Deze klas wordt gezien als de main van de applicatie. Heeft verschillende constanten voor zowel de foutmeldingen als de versie. En doet het volgende in de main:

- Er wordt een nieuwe style gemaakt.
- Maakt een nieuwe presentatie aan.
- Maakt ook een nieuwe SlideViewerFrame aan met de versie van de constante.
- Er wordt dan gebruik gemaakt van een try catch blok.
- Stel er worden geen argumenten gegeven, dan wordt de demo geladen.
- Als er wel argumenten zijn gegeven, dan wordt het XML-bestand geopend vanuit de argumenten.
- Stel er gaat iets fout dan wordt de exceptie gegooid met de foutmeldingen vanuit de constanten.

## KeyController

Maakt gebruik van de superclass KeyAdapter en heeft een field voor de presentatie.

- Bevat een constructor met de presentatie.
- Overschrijft een methode keyPressed vanuit de hoofdklas en gebruikt een switch case om te kijken welke toets er wordt ingedrukt.



## MenuController

Maakt gebruik van de superclass MenuBar.

- Bevat 2 fields, voor de huidige frame en voor de presentatie.
- Heeft een constante voor de serialVersionUID.
- Bevat heel veel constanten voor de namen van de knoppen.
- Bevat 2 constanten voor de bestandsnamen.
- Bevat 3 constanten voor foutmeldingen.

Daarnaast zijn er verschillende methodes zoals,

**De constructor**, slaat de gegeven frame en presentatie op, maakt een nieuwe menu voor de "file" tab. Voegt een nieuwe menu item toe met de mkMenuItem methode deze wordt opgeslagen in de MenuItem variabel om aanpassingen te doen. Aan deze MenuItem wordt een ActionListener toegevoegd die een aantal methodes doet gebaseerd op de soort menu. Dit wordt allemaal gedaan voor alle tabs en menu knoppen die in de constanten staan.

**MkMenuItem**, geeft een MenuItem terug met de gegeven naam en een shortcut van CTRL + eerste letter van de naam.

Dit kan een probleem opleveren omdat er dubbele shortcuts kan ontstaan, zoals de New en Next die allebei CTRL + N zijn.

## Presentation

- Bevat een field voor de volgende attributen, titel, een lijst met slides, de huidige slide nummer en de SlideViewerComponent.
- Bevat 2 verschillende constructors, één waarin de SlideViewerComponent leeg blijft en de clear methode wordt aangeroepen en de andere wordt de SlideViewerComponent wel meegegeven.

Daarnaast heeft de klasse nog een aantal methodes namelijk,

- GetSize, geeft de grootte van de slides lijst terug.
- SetSlideNumber, zet de currentSlideNumber naar het meegegeven getal.
- PrevSlide, gaat terug naar de vorige slide.
- NextSlide, gaat naar de volgende slide.
- Clear, maakt de slides lijst leeg.
- GetSlide, geeft de slide uit de lijst terug met de gegeven index.
- GetCurrentSlide, geeft de slide aan met de currentSlideNumber.
- Exit, beëindigt de applicatie met de gegeven exit code.

## Slide

- Bevat 2 constanten voor de breedte en de hoogte van de slide.
- Bevat een titel en een Vector voor de SlidelItems.
- Heeft een constructor om de lijst te initialiseren.

Heeft verschillende methodes en getters en setters,

- Append, voegt een SlidelItem toe aan de Vector. Als er een level en een bericht worden meegegeven wordt deze toegevoegd aan een nieuwe TextItem.

- getSlidelItem, geeft de SlidelItem terug van de gegeven index.

- getSize, geeft het aantal terug van de Vector lijst.

- draw, is een methode die de slide tekent er gebeuren een aantal dingen:

- Maakt een variabele aan voor de schaal met de getScale waarin de gegeven area Rectangle wordt meegegeven.

- Slaat de gegeven y positie op in een variabele.

- Maakt een textItem aan voor de titel.

- Pakt de stijl van de titel.

- Roept de draw methode aan van de SlidelItem met de gegeven parameters.

- Itereert door alle SlidelItems van deze Slide en doet dezelfde stappen.

- getScale, geeft het kleinste getal tussen de verhoudingen van de breedte en hoogte van de gegeven area tegenover de breedte en de hoogte van de slide.

## SlidelItem

Is een abstracte klasse, met de volgende fields en methodes,

- Heeft een field voor de level van de slidelItem.
- Bevat een constructor waar de level wordt meegegeven.
- Bevat een lege constructor met een level van 0.
- Heeft 2 abstracte methodes, getBoundingBox en draw.

## SlideViewerComponent

Maakt gebruik van de superclass JComponent.

- Heeft 4 fields voor, de huidige slide, het font voor de labels, een presentatie en een frame.
- Heeft een constante voor de serialVersionUID.
- Bevat constanten voor de stijl van de slide en de positie van de x en de y.
- Heeft een constructor die de achtergrondkleur instelt, een presentatie en frame initialiseren en een nieuw font aanmaakt.

Daarnaast zijn de volgende methodes te vinden in de klasse,

- GetPreferredSize, is overschreven uit de superclass en geeft een nieuwe Dimension terug met de breedte en hoogte van de Slide.
- Update, is overschreven uit de superclass en doet het volgende:
  - o Als er geen data is meegegeven dan wordt de repaint methode aangeroepen.
  - o Zet de presentatie en slide naar de gegeven presentatie en slide.
  - o Roept vanuit de superclass de repaint methode aan.
  - o Zet de titel van de frame naar de titel van de presentatie.
- PaintComponent, is overschreven uit de superclass en doet het volgende:
  - o Zet de achtergrond op de kleur uit de constante.
  - o Roept de fillRect methode aan voor de gegeven Graphics met de breedte en hoogte.
  - o Kijkt of er een slide is meegegeven als dit niet zo is dan wordt er returnt.
  - o Zet het font naar de opgeslagen font en de kleur naar de constante.
  - o Roept de drawString methode aan voor de Graphics om de tekst aan te geven hoever je in de presentatie bent, hierbij wordt er gebruik gemaakt van een placeholder.
  - o Maakt een nieuwe area aan met de ruimte van de component.
  - o Dan wordt de slide getekend.

## SlideViewerFrame

Maakt gebruik van de superclass JFrame.

- Heeft een constante voor de serialVersionUID.
- Bevat een constante voor de titel, breedte en hoogte.

Daarnaast zijn de volgende methodes te vinden in de klasse,

- Een constructor,
  - o Die roept de super aan met de gegeven titel.

- Maakt een nieuwe SlideViewerComponent aan met de gegeven presentatie.
- Zet de SlideViewerComponent van de presentatie naar de nieuwe SlideViewerComponent met gebruik van de setShowView methode.
- Roept de methode setupWindow aan met de SlideViewerComponent en presentation.
- SetupWindow,
  - Zet de titel naar de titel van de constante.
  - Voegt een nieuwe WindowListener toe om ervoor te zorgen dat het programma sluit met exit code 0 als de frame wordt gesloten.
  - Voegt de SlideViewerComponent toe aan de ContentPane.
  - Voegt een nieuwe KeyController toe aan de KeyListener.
  - Voegt een nieuwe MenuController toe aan de MenuBar.
  - Zet het formaat door een nieuwe Dimensie met de constante waardes.
  - Maakt de frame zichtbaar door de methode setVisible te gebruiken.

## Style

- Bevat een statische field met een array van Styles, waar de styles kan worden opgeslagen.
- Heeft een constante voor het lettertype.
- Heeft een indentatie, kleur, lettertype, lettertype grootte en de leading allemaal zonder acces modifier.
- Heeft de volgende methodes,
  - Een statische methode om 5 verschillende styles te maken. Deze worden in de styles array gezet.
  - Een statische methode om de style te krijgen op basis van de level.
  - Constructor die alle gegeven parameters opslaat in fields, waarbij er voor het lettertype een nieuwe wordt aangemaakt.
  - De getFont geeft een Font type terug op basis van de gegeven schaal.

## TextItem

### Maakt gebruik van de superclass Slideltem

- Bevat een field voor een tekst en heeft een constante als er geen tekst is gegeven.
- Heeft een constructor die een super aanroept en de tekst opslaat.
- Heeft een constructor met een lege textitem.

- Heeft een `getText` die of de opgeslagen tekst teruggeeft of een lege string als er niks is opgeslagen.
- Daarnaast zijn de volgende methodes te vinden in de klasse:
  - `GetAttributedString`, geeft een `AttributedString` terug en maakt een nieuwe `AttributedString` aan met de `getText` methode. Voegt daarnaast een `Attribute` toe met de font constante, de `getFont` methode en de index 0.
  - `GetBoundingBox`, deze methode wordt overgeschreven uit de superclass.
    - Maakt een lijst aan voor de `TextLayouts` vanuit de `getLayouts` methode.
    - Maakt een variabele aan voor de x en y formaat.
    - Itereerd met de `Iterator` door de layouts lijst.
    - Na de iteratie zijn de x en y formaten de volledige breedte en lengte van alle teksten.
    - Retournt een nieuwe `Rectangle` van de x keer de schaal, y is 0 en x en y formaat voor de lengte en de breedte.
  - `Draw` methode, deze methode wordt overschreven uit de superclass.
    - Wordt eerst gecontroleerd of de tekst leeg is of null.
    - Maakt een lijst voor de layouts en haalt het uit de `getLayouts` methode.
    - Maakt een nieuwe `Point` aan met de gegeven x en y en marges van de gegeven `Style`.
    - Er wordt een nieuwe `Graphics2D` aangemaakt, door de gegeven `Graphics` te casten.
    - Zet de kleur van de `Graphics`.
    - Itereerd door de layouts lijst heen.
    - En pakt dan de huidige layout en zet het in een variabele. Dan wordt de y positie veranderd met de `Ascent` van de `TextLayout`. Roept daarna de `draw` methode aan van de `TextLayout` met de gemaakte `Graphics` en de coördinaten van de `Point`. Veranderd de y positie van de `Point` met de `Descent` vanuit de `TextLayout`.
- `GetLayouts` methode, is een methode die gebruikt wordt in de bovenstaande methode. Er wordt een lijst teruggegeven met `TextLayouts`.
  - Eerst wordt er een nieuwe lijst gemaakt met `TextLayouts` die later wordt teruggegeven.
  - Dan wordt er een `AttributedString` opgeslagen vanuit de `getAttributedString` methode.
  - Maakt een nieuwe `Graphics2D` aan door de `Graphics` te casten.
  - Maakt daarna een nieuwe `FontRenderContext` aan, door gebruik te maken van de `getFontRenderContext`.

- Maakt een nieuwe LineBreakMeasurer aan met de AttributedString en de FontRenderContext.
- Berekent daarna de omslag van de breedte met de breedte constante van de Slide klas min de indentatie breedte van de gegeven style. Waarna dit wordt vermenigvuldigd met de schaal.
- Zolang de positie van de LineBreakMeasurer langer is dan de lengte van de tekst wordt er de volgende gedaan:
  - De volgende layout van de LineBreakMeasurer wordt opgehaald met de breedte die berekend is.
  - Deze layout wordt toegevoegd aan de layout lijst.

De lijst met layouts worden dan gereturt.

## XMLAccessor

Maakt gebruik van de superclass Accessor.

- Bevat constanten voor foutmeldingen en voor de standaard API die gebruikt wordt, maar deze wordt nergens voor gebruikt.
- Bevat heel veel constanten voor de XML-tags.
- Daarnaast worden de volgende methodes gebruikt in de klas,
  - GetTitle, verkrijgt een NodeList van de getElementsByTagName methode voor het gegeven element vanuit de parameter met de gegeven tag naam.
  - LoadFile, er worden eerst een aantal variabelen geïntialiseerd. Daarna wordt er een DocumentBuilder gemaakt met behulp van een DocumentBuilderFactory, Dan wordt er geprobeerd om het document te krijgen en de titel van de presentatie wordt ingesteld op basis van het document. Er wordt een nieuwe NodeList gemaakt voor de slides. Daarnaast wordt er een loop uitgevoerd over elke slide in de lijst:
    - Voor elke slide wordt er een nieuwe Slide gemaakt.
    - De titel van de slide wordt ingesteld door de methode getTitle aan te roepen.
    - Daarna wordt de slide aan de presentatie toegevoegd.
    - Een nieuwe lijst voor de items voor de slide wordt gemaakt.
    - Dan wordt er een nieuwe loop uitgevoerd voor elke item die in de lijst staat, voor elke item wordt de methode loadSlideItem aangeroepen om het item te laden.
- Dan kan er 3 excepties worden gegoooid, namelijk IOException, SAXException of ParserConfigurationException.

- LoadSlideItem, is een protected methode die de slides laadt met de gegeven slide en het item. Daarnaast wordt het volgende gedaan:
  - Er wordt een nieuwe level variabel aangemaakt met de waarde 1.
  - Dan worden de fields van het item opgehaald.
  - Ook wordt het level van uit de fields gehaald.
  - Dan wordt het type van het item wordt uit de fields gehaald.
  - Als het type een tekst is, wort er een nieuwe TextItem toegevoegd aan de slide.
  - Als het type een afbeelding is wordt er een nieuwe BitmapItem toegevoegd aan de slide.
  - Als het type onbekend is, wordt er een foutbericht meegegeven.
- SaveFile, kan een IOException gooien. En doet het volgende,
  - Er wordt een nieuwe PrintWriter aangemaakt met de bestandsnaam.
  - Er wordt XML geschreven met de showTitle van de presentatie.
  - Er wordt een loop uitgevoerd over elke slide in de presentatie.
    - Voor elke slide wordt dan de titel en een lijst van items naar het bestand geschreven.
    - Als het item van de slide een TextItem is wordt de level geprint en wordt de tekst uitgeprint. Vervolgens wordt hetzelfde gedaan met een BitmapItem, dan wordt ook de level geprint en de bitmapItem wordt opgehaald.
    - Stel het is een item van een onbekend type, dan wordt er een bericht naar de console afgedrukt.
- Uiteindelijk wordt de presentatie tag afgesloten.