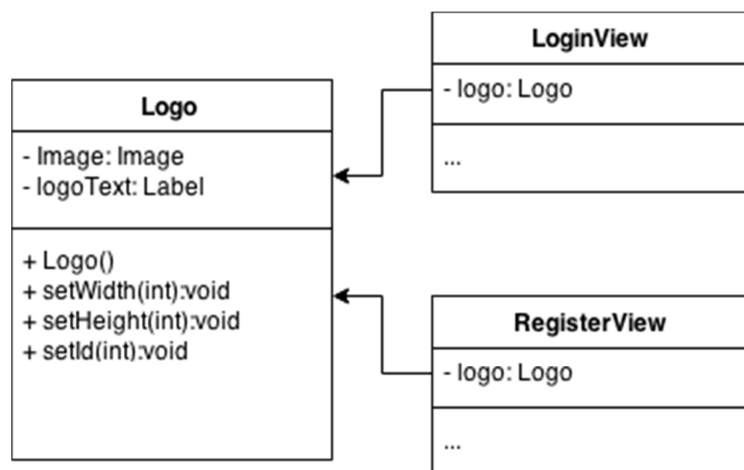


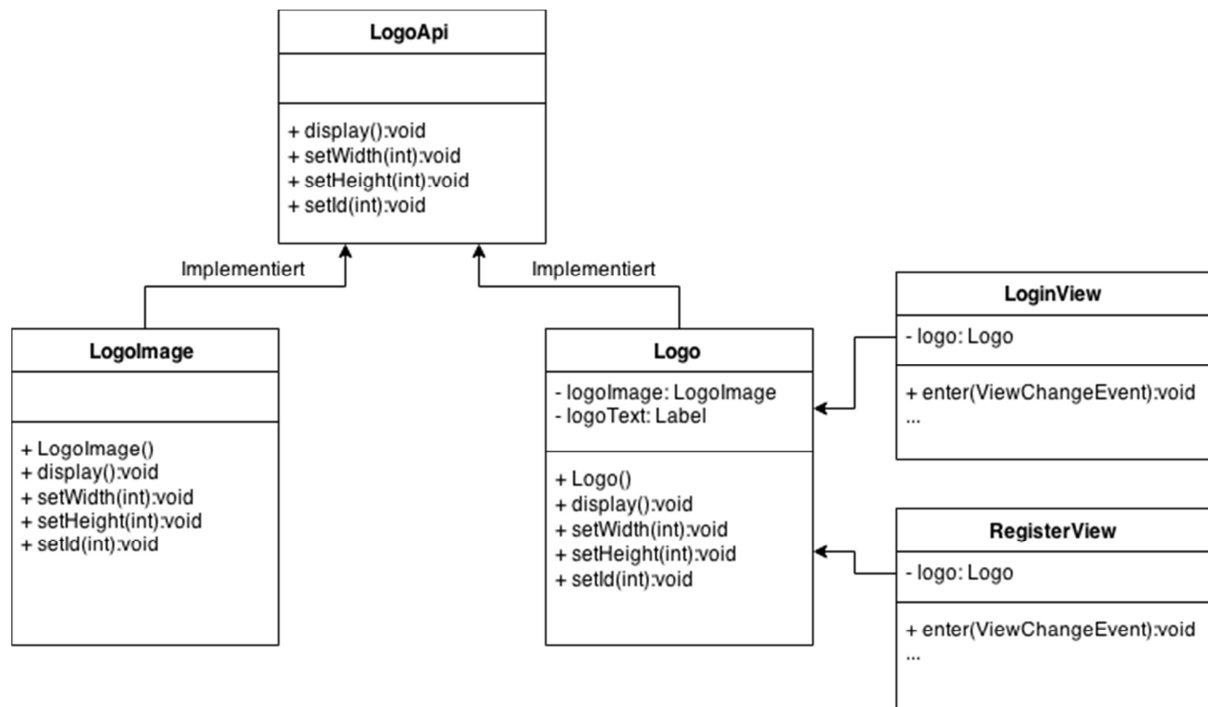
Design Pattern im Projekt SocialFunnel

Das Projekt verwendet nun das [Proxy-Pattern](#), um Ressourcen effizienter zu laden. Das Ziel des Proxy-Pattern ist es, Ressourcen nur wenn sie benötigt werden auch tatsächlich zu laden und ein mehrfaches Laden zu verhindern. Dieses Vorgehen soll den Speicherverbrauch reduzieren und Ladezeiten durch Wiederverwendung bereits geladener Ressourcen kürzen.

UML - vor Implementierung



UML - nach Implementierung



Code

LogoApi.java:

```

public interface LogoApi {

    public void display();
    public void setWidth(String width);
    public void setHeight(String height);
    public void setId(String id);

}

```

LogoImage.java:

```

3 import java.io.File;
8
9 public class LogoImage extends Image implements LogoApi{
10
11     public LogoImage() {
12         //empty image
13     }
14
15     @Override
16     public void display() {
17         if(this.getSource() == null){
18             if(null != VaadinService.getCurrent()){
19                 this.setSource(new FileResource(new File(
20                     VaadinService.getCurrent()
21                         .getBaseDirectory()
22                         .getAbsolutePath()
23                         + "/WEB-INF/images/logo.png")));
24             } else {
25                 this.setSource(new FileResource(new File(
26                     "/src/main/webapp/WEB-INF/images/logo.png")));
27             }
28             System.out.println("Bild geladen");
29
30             this.setWidth("300px");
31             this.setHeight("300px");
32         }
33     }
34
35     @Override
36     public void setHeight(String height) {
37         super.setHeight(height);
38     }
39
40     @Override
41     public void setId(String id) {

```

Logo.java:

```

@Override
public void display() {
    if(imgLogo==null)imgLogo = new LogoImage();
    imgLogo.display();
}

@Override
public void setHeight(String height) {
    if(imgLogo==null)imgLogo = new LogoImage();
    imgLogo.setHeight(height);
}

@Override
public void setId(String id) {
    if(imgLogo==null)imgLogo = new LogoImage();
    imgLogo.setId(id);
}

@Override
public void setWidth(String width) {
    if(imgLogo==null)imgLogo = new LogoImage();
    imgLogo.setWidth(width);
}

```

Aufruf in Login- und RegisterView.java:

```

@Override
public void enter(ViewChangeEvent event) {
    logo.display();
}

```

Testen des implementierten Patterns

Unit Tests machen Für dieses Pattern keinen Sinn, da es ja beabsichtigt ist das Verhalten der Klasse vollständig zu Kapseln und den Ladevorgang für Ressourcen zu optimieren. Dieses Verhalten lässt sich lediglich durch zusätzliche Ausgaben bei Laufzeit aufzeigen. Das Szenario in diesem Fall ist das Starten der Anwendung und mehrfaches hin und herspringen zwischen dem Login- und Registerview.

Dieses Vorgehen produziert folgende Ausgabe:

```

SocialFunnel - Application [Java Application] C:\interne\Programme\Jdk7u10\bin\javaw.exe
=====
2015-06-02 14:09:58.946 INFO 7256 --- [nio-8080-exec-1] c.vaadin.spr:
2015-06-02 14:09:58.950 INFO 7256 --- [nio-8080-exec-1] c.vaadin.spr:
Bild angefordert
Bild geladen
Bild angefordert
Bild geladen
Bild angefordert
Bild angefordert
Bild angefordert
Bild angefordert
Bild angefordert
Bild angefordert

```