



15. June 2016 by CodeLionX on Allgemein • Edit→

Final Unveiled – Fight against Injustice straightaway

Everyday a huge number of undetected crimes are committed on this planet. Whistleblowers try to impart those crimes to the community although politics and public authorities put rocks in their way. This project shall help these journalists and dedicated individuals to publish and save their captured video and photo material wherefore they have perhaps put their life at risk. Our Application addresses exactly this point.

Citizens with a conscience are not going to ignore wrong-doing simply because they'll be destroyed for it: the conscience forbids it.

Edward Snowden

People using our App can take pictures and videos which are immediately uploaded to our servers. There they are stored in a private library only accessible to the owner. Through a web based interface you are able to publish your

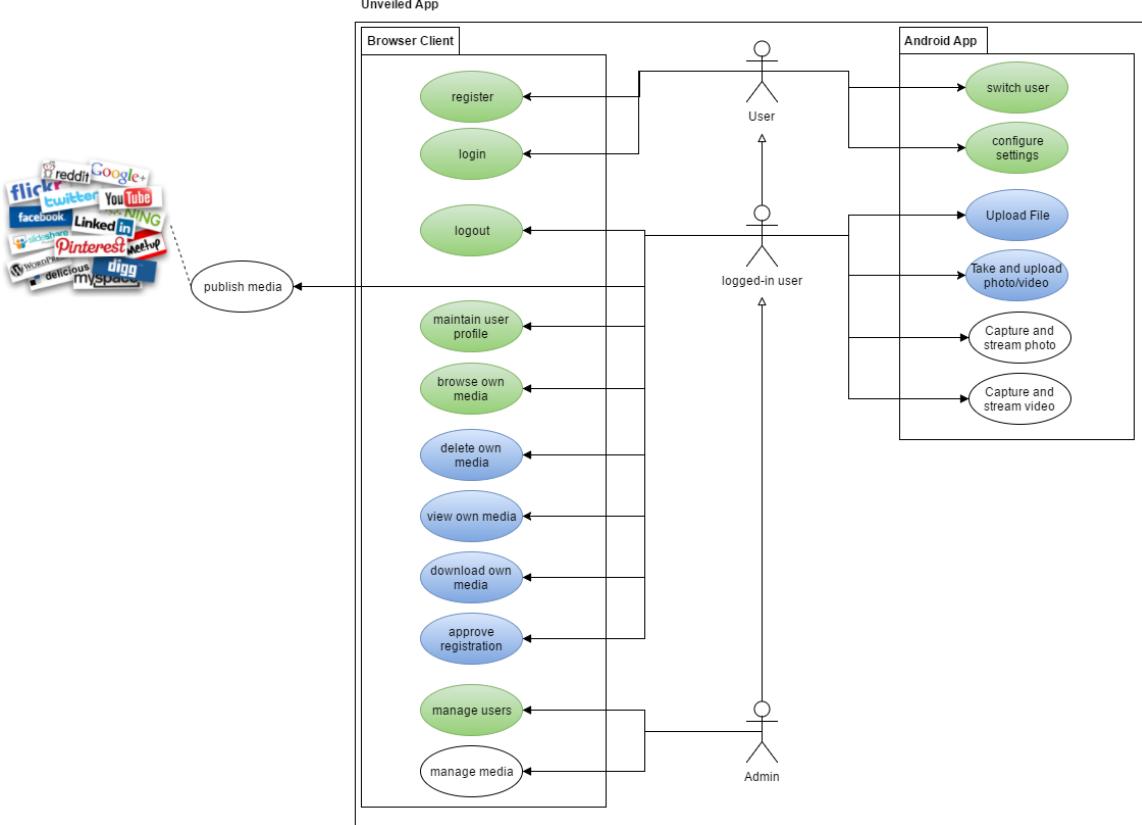
recorded content.

General

- [Homepage](#)
- [Blog](#)
- [Documents](#)
- [Source Code \(Github Organisation\)](#)
- [Agile Planning \(Jira\)](#)
- [Deployment & Test Logs \(Travis CI\)](#)
- Code Quality
 - [imflux – Sonarqube](#)
 - [imflux – Codacy](#)
 - [Unveiled-Server – Sonarqube](#)
 - [Unveiled-Server – Codacy](#)

Requirements Analysis

- Overall Use Case Diagram



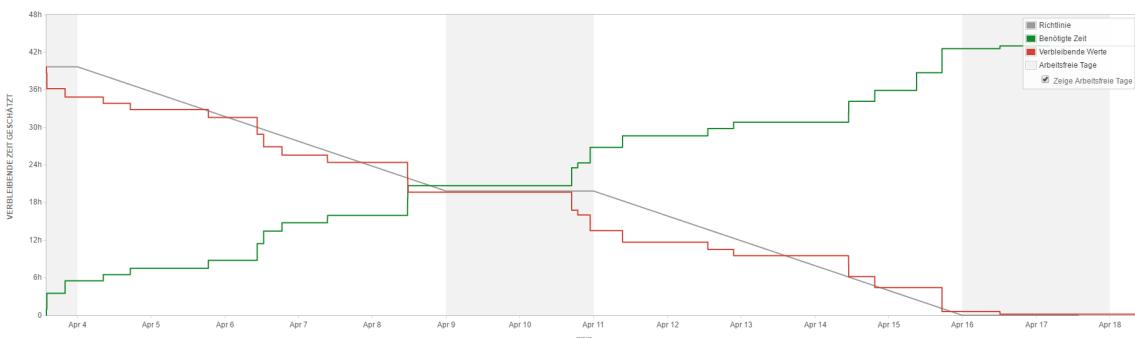
- Software Requirements Specification
 - Use Cases (with Functional Test Cases where applicable)
 - Use Case: Capture and stream video
 - Use Case: Configure settings
 - Use Case: Maintain user profile
 - Use Case: Switch user
 - Use Case: Register
 - Use Case: Browse media
 - Use Case: Manage Users
 - Use Case: Delete own Media
 - Use Case: Download own Media
 - Use Case: View own Media
 - Use Case: Approve Registrations
 - Use Case: Upload File
 - Test Plan
 - Blog-Posts:

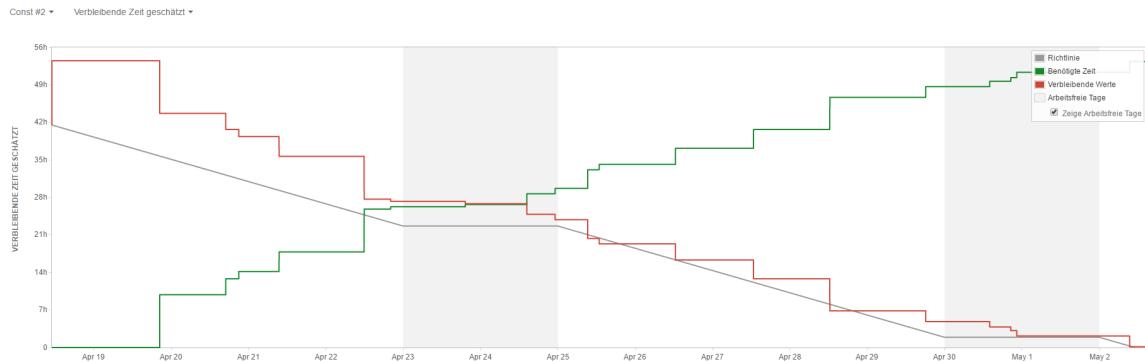
- Test Plan and Test Coverage
- Unit Testing
- Use Cases transformed into features
- Detailed Use Cases
- SRS and UCD
- Architecture and Team Roles

Project Planning

- Longterm Planning

@Ms. Berkling: We've discussed in class whether it's better to group the tasks by discipline or by phase first. We came to the decision that grouping the tasks by discipline first makes also sense.
- Term 1: Gantt-Chart, Team-Planning
- Term 2: Gantt-Chart, Team-Planning, Workload Estimation with FPs
- Agile Planning (Jira Board)
 - Two Healthy Burndown Charts:





- Function Point Calculation
- Blog-Posts:
 - FP calculation and time estimation (deprecated) – this is old stuff, consider the new document (Function Point Calculation) also mentioned above
 - GC RUP
 - Setup Jira Scrum Board

Architecture

- Software Architecture Document
- Metrics:
 - High Level Metrics:
 - imflux – Sonarqube
 - imflux – Codacy
 - Unveiled-Server – Sonarqube
 - Unveiled-Server – Codacy
 - Low Level Metrics:
 - imflux:

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method	
Number of Parameters (avg/max per method)	0.792	0.967	5	/imflux/src/main/java/sas/systems/imflux/session/r...		SingleParticipantSession	
Number of Static Attributes (avg/max per type)	49	0.79	1.893	11	/imflux/src/main/java/sas/systems/imflux/session/r...		
Efferent Coupling (avg/max per packageFragm)	4.357	3.637	11	/imflux/src/main/java/sas/systems/imflux/packet/tcp			
Specialization Index (avg/max per type)	0.139	0.452	3	/imflux/src/main/java/sas/systems/imflux/network/C...			
Number of Classes (avg/max per packageFragm)	62	4.429	3.812	13	/imflux/src/main/java/sas/systems/imflux/packet/tcp		
Number of Attributes (avg/max per type)	142	2.29	4.629	28	/imflux/src/main/java/sas/systems/imflux/session/r...		
Abstractness (avg/max per packageFragment)	0.23	0.336	1	/imflux/src/main/java/sas/systems/imflux/session			
Normalized Distance (avg/max per packageFragm)	0.339	0.367	1	/imflux/src/main/java/sas/systems/imflux/util			
Number of Methods (avg/max per type)	35	0.887	1.779	11	/imflux/src/main/java/sas/systems/imflux/packet/r...		
Number of Interfaces (avg/max per type)	14	1	1.464	4	/imflux/src/main/java/sas/systems/imflux/session/rp...		
Total Lines of Code	6048						
Weighted methods per Class (avg/max per typ	973	15.694	21.63	132	/imflux/src/main/java/sas/systems/imflux/session/r...		
Number of Methods (avg/max per type)	498	8.032	11.149	67	/imflux/src/main/java/sas/systems/imflux/session/r...		
Depth of Inheritance Tree (avg/max per type)	3	1.774	1.17	4	/imflux/src/main/java/sas/systems/imflux/network/D...		
Number of Packages	14						
Instability (avg/max per packageFragment)	0.529	0.39	1	/imflux/src/test/java/sas/systems/imflux/function...			
McCabe Cyclomatic Complexity (avg/max per	1.776	2.032	22	/imflux/src/main/java/sas/systems/imflux/session/r...		requestReceived	
junit	1.802	2.088	22	/imflux/src/main/java/sas/systems/imflux/session/r...		requestReceived	
sas.systems.imflux.session.tsp	2.343	3.641	22	/imflux/src/main/java/sas/systems/imflux/session/r...		requestReceived	
sas.systems.imflux.packet.tsp	2.326	3.609	10	/imflux/src/main/java/sas/systems/imflux/session/r...		requestReceived	
sas.systems.imflux.packet	1.818	2.066	12	/imflux/src/main/java/sas/systems/imflux/packet/pa...		encode	
sas.systems.imflux.session.rp	1.854	1.521	10	/imflux/src/main/java/sas/systems/imflux/session/r...		dataPacketReceived	
sas.systems.imflux.participant	1.477	1.161	9	/imflux/src/main/java/sas/systems/imflux/participan...		toString	
sas.systems.imflux.network	1.667	1.522	7	/imflux/src/main/java/sas/systems/imflux/network/C...		channelRead	
sas.systems.imflux.network.udp	1.391	1.093	6	/imflux/src/main/java/sas/systems/imflux/network/u...		decode	
sas.systems.imflux.util	1.438	0.704	3	/imflux/src/main/java/sas/systems/imflux/util/ByteL...		convertHexStringToByte...	
sas.systems.imflux.logging	1.167	0.373	2	/imflux/src/main/java/sas/systems/imflux/logging/L...		trace	
sas.systems.imflux.session	0	0					
java	1.594	1.582	9	/imflux/src/test/java/sas/systems/imflux/function...		testDeliveryToAllPartic...	
resources	0	0					
Nested Block Depth (avg/max per method)	1.4	0.46	5	/imflux/src/main/java/sas/systems/imflux/participan...		cleanup	
Lack of Cohesion of Methods (avg/max per ty	0.231	0.33	4	/imflux/src/main/java/sas/systems/imflux/session/r...			
Method Lines of Code (avg/max per method)	4235	7.911	13.371	125	/imflux/src/test/java/sas/systems/imflux/function...		testDeliveryToAllPartic...
Number of Overridden Methods (avg/max per	24	0.387	1.022	7	/imflux/src/main/java/sas/systems/imflux/session/r...		
Afferent Coupling (avg/max per packageFragm	7.643	9.232	27	/imflux/src/main/java/sas/systems/imflux/packet/tcp			
Number of Children (avg/max per type)	9	0.145	0.618	4	/imflux/src/main/java/sas/systems/imflux/packet/r...		

• Unveiled-Server:

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum	Method	
Number of Parameters (avg/max per method)	0.852	0.887	4	/Unveiled-Server/src/main/java/sas/systems/unveile...		initialize	
java	0.852	0.887	4	/Unveiled-Server/src/main/java/sas/systems/unveile...		initialize	
sas.systems.unveiled.server	1.5	0.957	4	/Unveiled-Server/src/main/java/sas/systems/unveile...		initialize	
sas.systems.unveiled.server.fileno	0.754	0.782	3	/Unveiled-Server/src/main/java/sas/systems/unveile...		FileWriter	
FileWriter.java	1.833	1.213	3	/Unveiled-Server/src/main/java/sas/systems/unveile...		FileWriter	
FileUploadServlet.java	1.083	0.64	2	/Unveiled-Server/src/main/java/sas/systems/unveile...		doPost	
FilePOJO.java	0.516	0.561	2	/Unveiled-Server/src/main/java/sas/systems/unveile...		FilePOJO	
FileParameters.java	0.5	0.5	1	/Unveiled-Server/src/main/java/sas/systems/unveile...		FileParameters	
sas.systems.unveiled.server.util	0.292	0.455	1	/Unveiled-Server/src/main/java/sas/systems/unveile...		loadPropertiesFile	
resources	0	0					
Nested Block Depth (avg/max per method)	1.4	0.46	5	/Unveiled-Server/src/main/java/sas/systems/unveile...			
Lack of Cohesion of Methods (avg/max per ty	0.231	0.33	4	/Unveiled-Server/src/main/java/sas/systems/unveile...			
Method Lines of Code (avg/max per method)	4235	7.911	13.371	125	/Unveiled-Server/src/test/java/sas/systems/unveile...		
Number of Overridden Methods (avg/max per	24	0.387	1.022	7	/Unveiled-Server/src/main/java/sas/systems/unveile...		
Afferent Coupling (avg/max per packageFragm	7.643	9.232	27	/Unveiled-Server/src/main/java/sas/systems/unveile...			
Number of Children (avg/max per type)	9	0.145	0.618	4	/Unveiled-Server/src/main/java/sas/systems/unveile...		

- Risk Table
- Continous Deployment

✓ SAS-Systems/Unveiled-Server
23

⌚ Duration: 2 min 29 sec
📅 Finished: 2 days ago

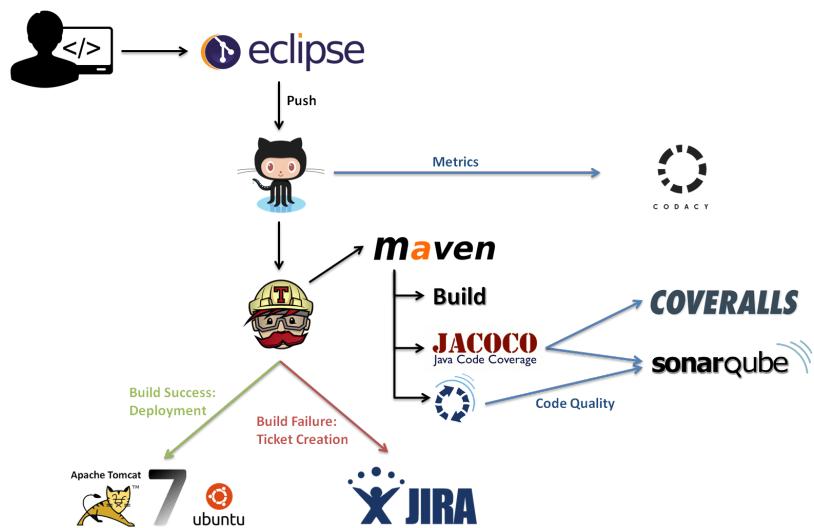
✓ SAS-Systems/Unveiled
138

⌚ Duration: 11 min 59 sec
📅 Finished: 3 days ago

✓ SAS-Systems/imflux
50

⌚ Duration: 3 min 10 sec
📅 Finished: 6 days ago

- Travis CI Logs (Build, Test, Deployment,)
 - [Unveiled](#)
 - [Unveiled-Server](#)
 - [imflux](#)
- Build / Deployment Lifecycle



- Blog-Posts:
 - [Automatic Deployment](#)
 - [Metrics](#)
 - [Implement Design Pattern in Project Code](#)
 - [2nd term Scope and Risk Management](#)
 - [MVC](#)
 - [Class Diagram](#)
 - [Architecture and Team Roles](#)

Demo

- [Webinterface](#) (Optimized for the use with Google Chrome)
- Unveiled Android App
 - [Installation Guide](#)

- APK-Download (You will need an Android Device to install and run the App)
- Source Code (Github Organisation)
 - Unveiled Android App
 - Unveiled Webinterface and (PHP-)Backend
 - Unveiled-Server (Java)
 - imflux – RTP streaming library
- Blog-Posts:
 - Installation
 - Backend API Documentation

Presentations

- Midterm Presentation
- Midterm Handout
- Final Presentation
- Final Handout



10. June 2016 by CodeLionX on Allgemein • Edit→

HW20: Installation

Dear Blog-Readers,

today we would like you to take action. We have build our Android App as an APK and published it on Github.

Please follow this guide to install our Android App on your device. We would really appreciate if you can note your

actions and results. You can use the following table for that:

Executor	name		
Date	date		
Device Model	device name	Android Version	version
Signup successful:		X	
Comments:			
Security Settings Change successful:			
Comments:			
Download successful:			
Comments:			
Application start successful:			
Comments:			
Login successful:			
Comments:			
Tested Application successfully:			
Comments:			

Please submit the table to us via our comment functionality or via email to unveiled@gmx.de, so that we can analyse the test result.

Regards

CodeLionX

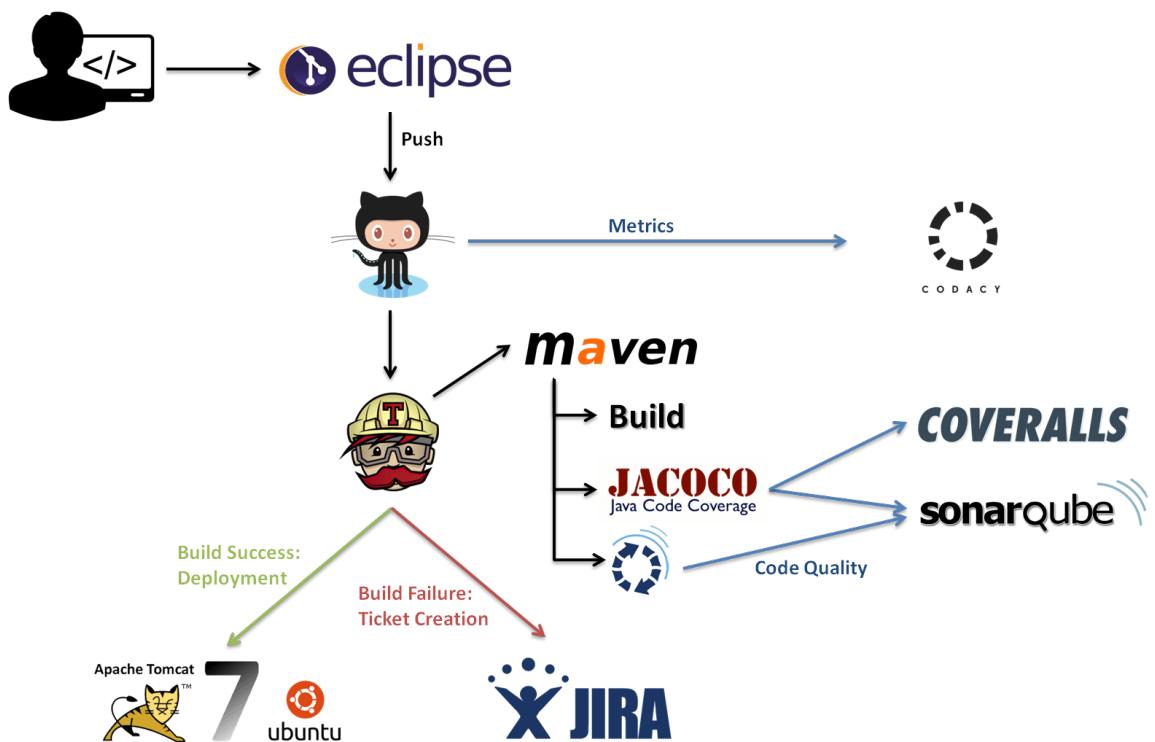


30. May 2016 by CodeLionX on Allgemein • Edit→

HW19: Automatic Deployment

Hi together,

today we want to describe you our deployment process. We used [TravisCI](#) for Continuous Integration and you can find our build of our streaming library imflux [here](#) and for our Java-Backend-Stack [here](#). The following picture shows our build lifecycle and the tool chain we use for our Continuous Integration (Continuous Delivery).



We use the following tool chain (for the Unveiled-Server build):

- `mvn install -DskipTests=true -Dmaven.javadoc.skip=true -B -V`

We use Maven for dependency management and building our application.

- `mvn clean test jacoco:report`

JaCoCo is used for running JUnit tests and collecting the test results.

- If the build was successful following things are done:

- `mvn coveralls:report`

The test results are reported to coveralls.io to make them available for analysis (currently there are no test for the Unveiled-Server build, but you can check the [imflux test results](#)).

- `mvn sonar:sonar -Dsonar.host.url=$SONAR_HOST_URL`

Sonarqube analyses the binary files and reports the results to [the website](#).

- `mvn tomcat7:deploy -Ddeploy.url=$DEPLOY_URL -Ddeploy.username=$DEPLOY_USER -Ddeploy.password=$DEPLOY_PASSWORD`

The builded .war file is deployed to our [own server](#) using the tomcat7 maven plugin.

- `./send_jira_ticket.sh`

If the build was not successful a new Jira ticket is created and sent to the [Jira server](#). Therefore we use a short shell script using curl to send a Json to the Jira API. You can see the result in the following picture:

The screenshot shows a Jira ticket for a Travis build error. The ticket is titled "Travis Build Error: deploy". It has the following details:

- Typ:** Bug
- Priorität:** Medium
- Komponente(n):** Keine
- Stichwörter:** Keine
- Status:** FERTIG (Arbeitsablauf anzeigen)
- Lösung:** Fertig

Beschreibung: The build process of commit: 4ded9b802f7114b3c20c1e9bb32816b6b7bfaa9c was not successful. Please visit <https://travis-ci.org/SAS-Systems/imflux/builds/133949222> This information was automatically created. Please add further instructions.

Aktivität:

- Allgemein
- Kommentare** (selected)
- Arbeitsprotokoll
- Änderungshistorie
- Aktivität

Sebastian Schmidl hat einen Kommentar hinzugefügt - in 1 Minute
just a CI test, can be safely deleted

All this information can be found in our travis.yml of the repositories imflux and Unveiled-Server.

Have a nice week

CodeLionX



30. May 2016 by CodeLionX on Allgemein • Edit→

HW18: Metrics

Hey folks,

today we want to share with you, how code metrics helped us to improve our code quality. We have done the metrics analysis within our eclipse IDE and the tool Metrics 1.3.6. Metrics 1.3.6 allows us to run low level analysis. Unfortunately we were not able to find a comparable tool which allows this low level analysis within our build lifecycle. But you can take a look at our Sonarqube projects

([imflux](#), [Unveiled-Server](#)). Sonarqube shows some high level metrics.

The following picture shows the results of Metrics 1.3.6 of the Unveiled-Server project before our changes.

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
▷ Number of Parameters (avg/max per method)		1,02	1,769	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO
▷ Number of Static Attributes (avg/max per type)	30	2,308	1,488	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Efferent Coupling (avg/max per packageFragment)		3	0,816	4	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Specialization Index (avg/max per type)		0,154	0,411	1,5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Classes (avg/max per packageFragment)	13	4,333	1,247	6	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Attributes (avg/max per type)	50	3,846	4,671	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Abstractness (avg/max per packageFragment)		0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Normalized Distance (avg/max per packageFragment)		0,475	0,195	0,625	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Static Methods (avg/max per type)	2	0,154	0,361	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Interfaces (avg/max per packageFragment)		0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Total Lines of Code	1033					
▷ Weighted methods per Class (avg/max per type)	162	12,462	10,66	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Methods (avg/max per type)	97	7,462	8,828	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Depth of Inheritance Tree (avg/max per type)			1,308	0,722	3	/Unveiled-Server/src/main/java/sas/systems/unveile...
▷ Number of Packages	3					
▷ Instability (avg/max per packageFragment)		0,525	0,195	0,8	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ McCabe Cyclomatic Complexity (avg/max per type)	1,636	1,743		15	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
▷ java		1,636	1,743	15	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
▷ sas.systems.unveiled.server.fileio		1,533	2,115	15	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
▷ FileUploadServlet.java		3,667	5,121	15	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
▷ FileWriter.java		2,333	0,745	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	writeToFile
▷ FilePOJO.java		1	0	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO
▷ sas.systems.unveiled.server		1,8	1,579	8	/Unveiled-Server/src/main/java/sas/systems/unveile...	announceRequestReceived
▷ sas.systems.unveiled.server.util		1,625	0,992	4	/Unveiled-Server/src/main/java/sas/systems/unveile...	SessionManager
▷ resources		0	0			
▷ Nested Block Depth (avg/max per method)		1,323	0,664	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	optionsRequestReceived
▷ Lack of Cohesion of Methods (avg/max per type)		0,383	0,396	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Method Lines of Code (avg/max per method)	572	5,778	10,64	84	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
▷ Number of Overridden Methods (avg/max per type)	2	0,154	0,361	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Afferent Coupling (avg/max per packageFragment)		3	1,633	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ Number of Children (avg/max per type)	0	0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	

As you can see we have two possible points where we are not in the green (in this case blue) area. The first one is the number of parameters, where we have a maximum of 16 parameters per method. This method is a constructor of our File POJO class. We decided to not change this class, because it represents a database record. Instead we decided to change the Cyclomatic Complexity of our `doPost()` method our `FileUploadServlet` class. You can find the code before the refactoring in our [Github-repository](#).

To improve our code quality and the cyclomatic complexity of the `doPost()` method, we extracted some logic into new methods. Therefore it was necessary to create a new class:

FileParameters and a new exception

class: BadRequestException:

```
/**  
 * Exception class for the parameter parsing method  
 *  
 * @author CodeLionX  
 */  
private class BadRequestException extends Exception {  
  
    private static final long serialVersionUID = 1L;  
  
    public BadRequestException(String string)  
        super(string);  
    }  
}
```

You can find the new source code [here](#). The following screenshot shows you the metrics analysis result after refactoring:

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
▷ Number of Parameters (avg/max per method)	1,019	1,716	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
▷ Number of Static Attributes (avg/max per type)	31	2,067	1,526	5 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Efferent Coupling (avg/max per packageFragment)		3	0,816	4 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Specialization Index (avg/max per type)		0,118	0,375	1,5 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Classes (avg/max per packageFragment)	15	5	0,816	6 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Attributes (avg/max per type)	51	3,4	4,499	16 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Abstractness (avg/max per packageFragment)		0	0	0 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Normalized Distance (avg/max per packageFragment)		0,475	0,195	0,625 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Static Methods (avg/max per type)	2	0,133	0,34	1 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Interfaces (avg/max per packageFragment)	0	0	0	0 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Total Lines of Code	1053					
▷ Weighted methods per Class (avg/max per type)	167	11,133	10,893	33 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Methods (avg/max per type)	104	6,933	8,575	33 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Depth of Inheritance Tree (avg/max per type)			1,4	0,8 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Packages	3					
▷ Instability (avg/max per packageFragment)		0,525	0,195	0,8 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ McCabe Cyclomatic Complexity (avg/max per type)		1,575	1,181	8 /Unveiled-Server/src/main/java/sas/systems/unveile...	announceRequestReceived	
▷ java		1,575	1,181	8 /Unveiled-Server/src/main/java/sas/systems/unveile...	announceRequestReceived	
▷ sas.systems.unveiled.server		1,8	1,579	8 /Unveiled-Server/src/main/java/sas/systems/unveile...	announceRequestReceived	
▷ sas.systems.unveiled.server.fileio		1,423	0,948	6 /Unveiled-Server/src/main/java/sas/systems/unveile...	doPost	
▷ FileUploadServlet.java		2,167	1,462	6 /Unveiled-Server/src/main/java/sas/systems/unveile...	doPost	
▷ FileUploadServlet		2,273	1,483	6 /Unveiled-Server/src/main/java/sas/systems/unveile...	doPost	
doPost	6					
readRequest	3					
authenticateUserWithToken	3					
getDoubleSilently	3					
getIntSilently	3					
getBooleanSilently	2					
FileUploadServlet	1					
init	1					
destroy	1					
createDbEntry	1					
writeFile	1					
▷ BadRequestException		1	0	1 /Unveiled-Server/src/main/java/sas/systems/unveile...	BadRequestException	
▷ FileWriter.java		2,333	0,745	3 /Unveiled-Server/src/main/java/sas/systems/unveile...	writeToFile	
▷ FileParameters.java		1	0	1 /Unveiled-Server/src/main/java/sas/systems/unveile...	FileParameters	
▷ FilePOJO.java		1	0	1 /Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
▷ sas.systems.unveiled.server.util		1,625	0,992	4 /Unveiled-Server/src/main/java/sas/systems/unveile...	SessionManager	
resources	0	0				
▷ Nested Block Depth (avg/max per method)	1,34	0,671	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	optionsRequestReceived	
▷ Lack of Cohesion of Methods (avg/max per type)	0,337	0,395	1	/Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Method Lines of Code (avg/max per method)	596	5,623	7,717	37 /Unveiled-Server/src/main/java/sas/systems/unveile...	insertFile	
▷ Number of Overridden Methods (avg/max per type)	2	0,133	0,34	1 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Afferent Coupling (avg/max per packageFragment)		3	1,633	5 /Unveiled-Server/src/main/java/sas/systems/unveile...		
▷ Number of Children (avg/max per type)	0	0	0	0 /Unveiled-Server/src/main/java/sas/systems/unveile...		

So you can see that we now have reduced the cyclomatic complexity of the `doPost()` method from 15 to 6 by introducing six new methods and two new classes.

EDIT:

As it turned out that we had to optimize two different aspects, we have done that later on:

As we only have 2 metrics that are not good, we have also optimized the one we stated before as not necessary. This is the Metric *Number of Parameters*, which is bad for our `FilePOJO` java class. See the screenshots below:

Screenshot before the second refactoring:

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Parameters (avg/max per method)	0,974	1,646	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
java	0,974	1,646	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
sas.systems.unveiled.server.fileio	0,984	2,051	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
▷ FilePOJO.java	0,97	2,702	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO	
▷ FileWriter.java	1,833	1,213	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	FileWriter	
▷ FileUploadServlet.java	1,083	0,64	2	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost	
▷ FileParameters.java	0,5	0,5	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	FileParameters	
▷ sas.systems.unveiled.server	1,5	0,957	4	/Unveiled-Server/src/main/java/sas/systems/unveile...	initialize	
▷ sas.systems.unveiled.server.util	0,292	0,455	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	loadPropertiesFile	
resources	0	0				
Number of Static Attributes (avg/max per type)	36	2,4	1,583	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Efferent Coupling (avg/max per packageFragment)		3	0,816	4	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Specialization Index (avg/max per type)		0,118	0,375	1,5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Classes (avg/max per packageFragment)	15	5	0,816	6	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Attributes (avg/max per type)	52	3,467	4,47	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Abstractness (avg/max per packageFragment)		0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Normalized Distance (avg/max per packageFragment)		0,475	0,195	0,625	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Static Methods (avg/max per type)	2	0,133	0,34	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Interfaces (avg/max per packageFragment)	0	0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Total Lines of Code	1106					
Weighted methods per Class (avg/max per type)	179	11,933	10,554	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Methods (avg/max per type)	115	7,667	8,506	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
java	115	7,667	8,506	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
sas.systems.unveiled.server.fileio	63	12,6	10,929	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ FilePOJO.java	33	33	0	33	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ FileParameters.java	12	12	0	12	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ FileUploadServlet.java	12	6	5	11	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ FileWriter.java	6	6	0	6	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ sas.systems.unveiled.server	29	7,25	5,932	17	/Unveiled-Server/src/main/java/sas/systems/unveile...	
▷ sas.systems.unveiled.server.util	23	3,833	4,776	14	/Unveiled-Server/src/main/java/sas/systems/unveile...	
resources	0	0	0			
Depth of Inheritance Tree (avg/max per type)		1,4	0,8	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Packages	3					
Instability (avg/max per packageFragment)		0,525	0,195	0,8	/Unveiled-Server/src/main/java/sas/systems/unveile...	
McCabe Cyclomatic Complexity (avg/max per type)		1,53	1,137	8	/Unveiled-Server/src/main/java/sas/systems/unveile...	announceRequestReceived
Nested Block Depth (avg/max per method)		1,316	0,649	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	optionsRequestReceived
Lack of Cohesion of Methods (avg/max per type)		0,37	0,386	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Method Lines of Code (avg/max per method)	620	5,299	7,454	37	/Unveiled-Server/src/main/java/sas/systems/unveile...	insertFile
Number of Overridden Methods (avg/max per type)	2	0,133	0,34	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Afferent Coupling (avg/max per packageFragment)		3	1,633	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Children (avg/max per type)	0	0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	

I changed the constructor to only get two parameters and set the other attributes to default values. You can see all the changes [here](#).

And this is the result of the metrics tool after the second refactoring:

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
Number of Parameters (avg/max per method)	0,852	0,887		4	/Unveiled-Server/src/main/java/sas/systems/unveile...	initialize
java	0,852	0,887		4	/Unveiled-Server/src/main/java/sas/systems/unveile...	initialize
sas.systems.unveiled.server	1,5	0,957		4	/Unveiled-Server/src/main/java/sas/systems/unveile...	initialize
sas.systems.unveiled.server.fileio	0,754	0,782		3	/Unveiled-Server/src/main/java/sas/systems/unveile...	FileWriter
FileWriter.java	1,833	1,213		3	/Unveiled-Server/src/main/java/sas/systems/unveile...	FileWriter
FileUploadServlet.java	1,083	0,64		2	/Unveiled-Server/src/main/java/sas/systems/unveile...	doPost
FilePOJO.java	0,516	0,561		2	/Unveiled-Server/src/main/java/sas/systems/unveile...	FilePOJO
FileParameters.java	0,5	0,5		1	/Unveiled-Server/src/main/java/sas/systems/unveile...	FileParameters
sas.systems.unveiled.server.util	0,292	0,455		1	/Unveiled-Server/src/main/java/sas/systems/unveile...	loadPropertiesFile
resources	0	0				
Number of Static Attributes (avg/max per type)	36	2,4	1,583	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Efferent Coupling (avg/max per packageFragment)		3	0,816	4	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Specialization Index (avg/max per type)		0,118	0,375	1,5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Classes (avg/max per packageFragment)	15	5	0,816	6	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Attributes (avg/max per type)	52	3,467	4,47	16	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Abstractionness (avg/max per packageFragment)		0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Normalized Distance (avg/max per packageFragment)		0,475	0,195	0,625	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Static Methods (avg/max per type)	2	0,133	0,34	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Interfaces (avg/max per packageFragment)	0	0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Total Lines of Code	1107					
Weighted methods per Class (avg/max per type)	177	11,8	10,297	31	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Methods (avg/max per type)	113	7,533	8,115	31	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Depth of Inheritance Tree (avg/max per type)			1,4	0,8	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Packages	3					
Instability (avg/max per packageFragment)		0,525	0,195	0,8	/Unveiled-Server/src/main/java/sas/systems/unveile...	
McCabe Cyclomatic Complexity (avg/max per packageFragment)		1,539	1,144	8	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Nested Block Depth (avg/max per method)		1,322	0,653	3	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Lack of Cohesion of Methods (avg/max per type)		0,37	0,386	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Method Lines of Code (avg/max per method)	629	5,47	7,674	37	/Unveiled-Server/src/main/java/sas/systems/unveile...	insertFile
Number of Overridden Methods (avg/max per type)	2	0,133	0,34	1	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Afferent Coupling (avg/max per packageFragment)		3	1,633	5	/Unveiled-Server/src/main/java/sas/systems/unveile...	
Number of Children (avg/max per type)	0	0	0	0	/Unveiled-Server/src/main/java/sas/systems/unveile...	

As you can see, we improved the maximum of the metric *Number of Parameters* from 16 to 4.

Have a nice week

CodeLionX



22. May 2016 by CodeLionX on Allgemein • Edit→

HW17: Test plan and coverage

Hi together,

today we want to provide you the link to our testplan. This document describes all the test we are doing to ensure we

have no bugs in our new implemented features. You can find it in under [documentation](#).

Additionally we also reached our goal to have more than 50% test coverage for our streaming library imflux. You can find all test reports on [coveralls](#) or on [sonarqube](#) (Please be aware that you might not be able to use this link inside the DHBW network. Please use [this link](#) instead.).

Have a nice week

CodeLionX



8. May 2016 by CodeLionX on Allgemein • Edit→

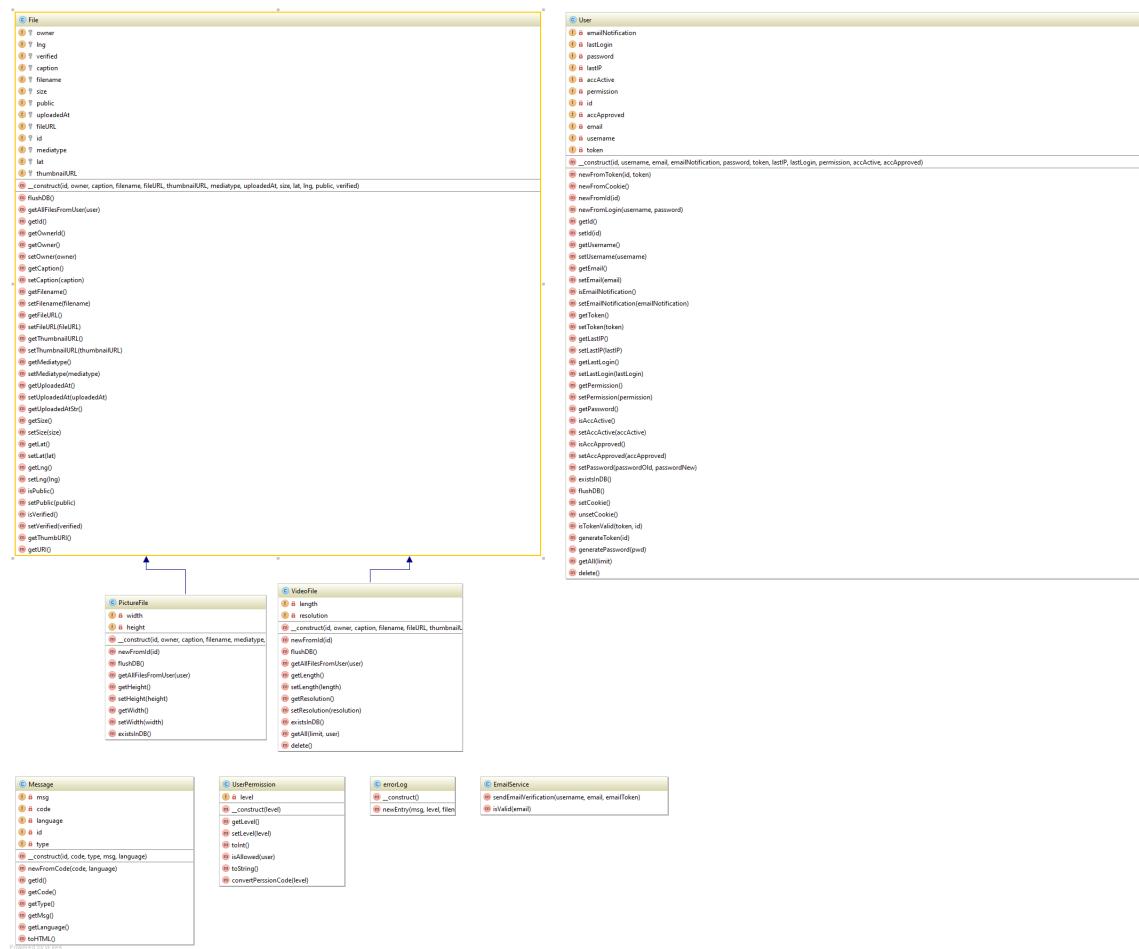
HW16: Implement Design Pattern in Project Code

Hi together,

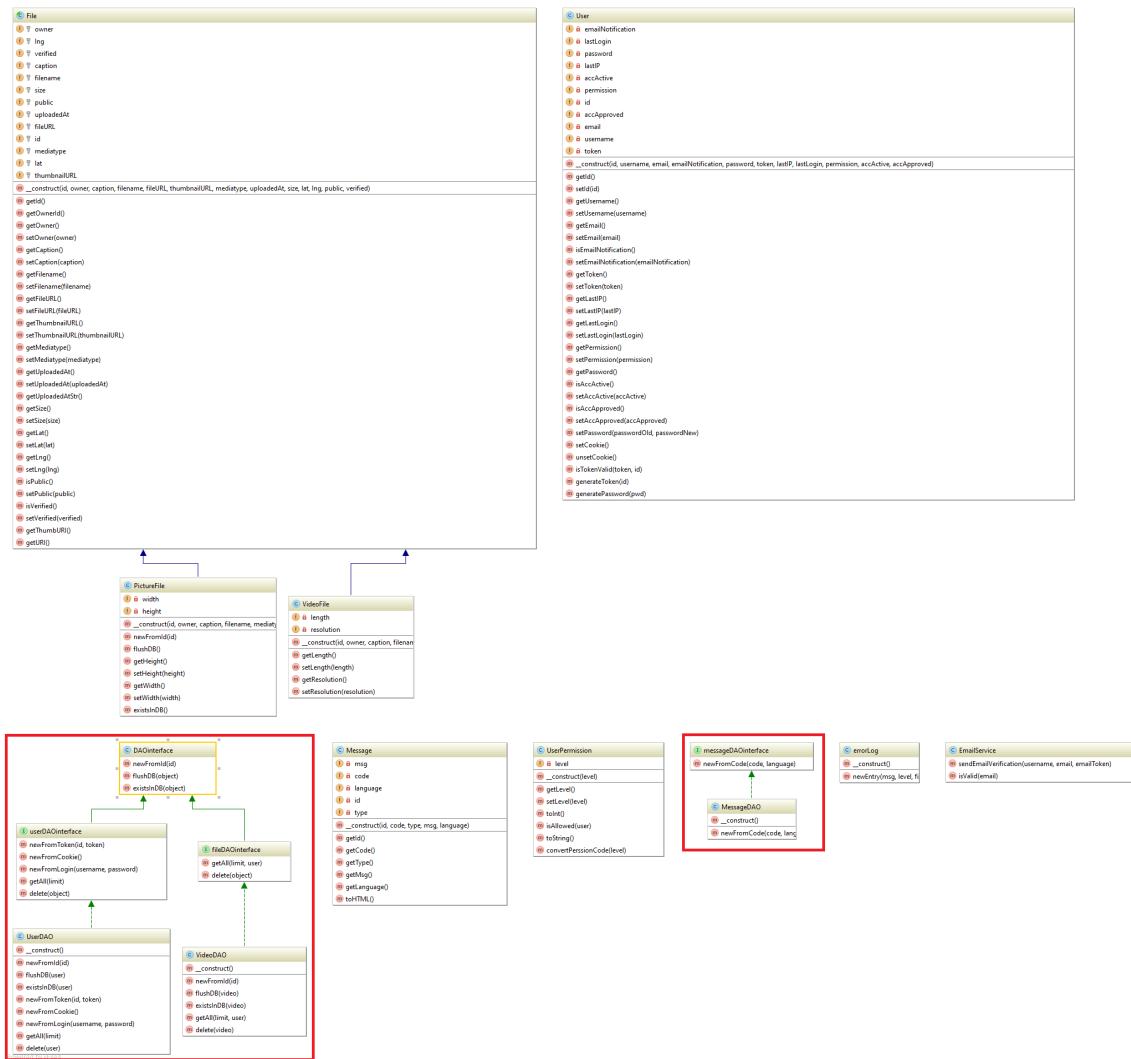
this weeks task was to refactor a part of our project code to implement a Design Pattern. We have chosen to implement the Data Access Object Pattern (DAO Pattern) in our server-side code. We have used it to encapsulate our low-level database communication from our server logic. The DAO Pattern is made up of

- *Model Objects* which represent one database entry,
 - A *Data Access Object Interface (DAO Interface)* which defines standard actions to be performed on the *Model Objects* and
 - the *Data Access Object* itself which implements the *DAO Interface* and is responsible for the communication with the data storage.

The following screenshot shows our **old** implementation of our Backend logic:



After refactoring our code the class diagram looked like this (the **new** implementation):



You can click on both pictures to open them in full resolution.

Have a nice Sunday,

team Unveiled



28. April 2016 by CodeLionX on Allgemein • Edit→

HW15: Fowler

refactoring

Hey there,

this weeks homework was to work through chapter 1 of Fowler's book *Refactoring* and doing the steps he did in the example. We would like to share our Github-Repos in which we documented our refactoring steps. Use the following links:

- CodeLionX: [Fowler/refactor](#)
- SeanAda: [Fowler/master](#)
- Digister: [Fowler-Software-Engineering-Homework/master](#)

We hope you like our work and wish you a nice rest of the week.

Greetings

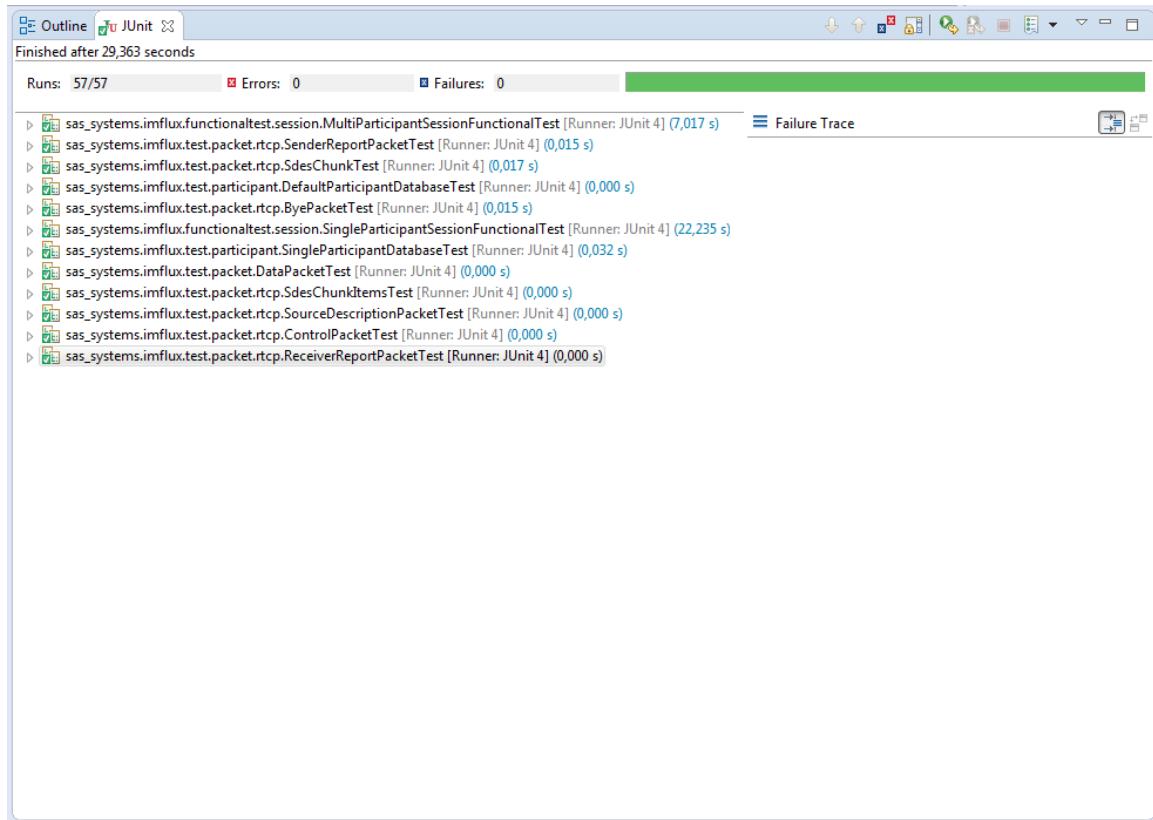
– team Unveiled



25. April 2016 by CodeLionX on Allgemein • Edit→

HW14: Unittesting

With this blog entry we want to share our testing approach. We are using JUnit for testing our backend. You can find our [Testing code](#) on Github and the following screenshot shows our IDE running these tests:



We are also using [Travis-CI](#) and [Coveralls](#) for running our tests automatically. Our build-tool is Maven and we use it to perform our testing on Travis. You can find our Maven build-file [here](#). See the below screenshots:

```

1528 [INFO] Starting Coveralls job for travis-ci (124033212)
1529 [INFO] Git commit 2ea024e in deploy
1530 [INFO] Writing Coveralls data to /home/travis/build/SAS-Systems/imflux/target/coveralls.json...
1531 [INFO] Processing coverage report from /home/travis/build/SAS-Systems/imflux/target/site/jacoco/jacoco.xml
1532 [INFO] Successfully wrote Coveralls data in 525ms
1533 [INFO] Gathered code coverage metrics for 42 source files with 8269 lines of code:
1534 [INFO] - 2264 relevant lines
1535 [INFO] - 1194 covered lines
1536 [INFO] - 1070 missed lines
1537 [INFO] Submitting Coveralls data to API
1538 [INFO] Successfully submitted Coveralls data in 848ms for Job #28.1
1539 [INFO] https://coveralls.io/jobs/13689600
1540 [INFO] *** It might take hours for Coveralls to update the actual coverage numbers for a job
1541 [INFO] If you see question marks in the report, please be patient
1542 [INFO] -----
1543 [INFO] BUILD SUCCESS
1544 [INFO] -----
1545 [INFO] Total time: 46.180 s
1546 [INFO] Finished at: 2016-04-18T21:48:50+00:00
1547 [INFO] Final Memory: 27M/491M
1548 [INFO] -----
1549
1550
1551 Done. Your build exited with 0.

```

The screenshot shows the GitHub repository page for SAS-Systems / IMFLUX. At the top right, it displays a coverage percentage of 54%. Below the header, there are navigation links for 'BRANCH: MASTER', 'NOTIFICATIONS', 'CHANGE SOURCE', and 'GITHUB REPO'. The main section is titled 'LATEST BUILDS' and lists seven recent builds:

BUILD	BRANCH	COVERAGE	COMMENT	COMMITTER	TYPE	TIME	VIA
#28	deploy	52.74	created test for single participant session	 CodeLionX	push	2 days ago	travis-ci
#27	deploy	50.96	improved single participant database test	 CodeLionX	push	4 days ago	travis-ci
#26	deploy	50.27	running test with half network load: just one packet per session-session connection	 CodeLionX	push	4 days ago	travis-ci
#21	master	53.75	Merge pull request #5 from SAS-Systems/coverallsTest set up coveralls	 CodeLionX	push	11 Apr 2016	travis-ci
#20	master	53.75	next test	 CodeLionX	PULL #5	11 Apr 2016	travis-ci
#19	coverallsTest	53.75	next test	 CodeLionX	push	11 Apr 2016	travis-ci

Demo Application in TDD

First of all we used Node.js to install the Mocha and Chai javascript Framework for testing. Now that we have access to the frameworks we were able to write our first tests and implement the function afterwards. For Node.js there is no need of an IDE, therefore we just used Sublime Text 2 and the command console.

An advantage of Mocha and Chai is that the test is really easy to read. For example:

```
describe('calculator', function(){
    it('should add two numbers', function(){
        var result = calculate('3', '+', '5')
        expect(result).to.equal(8)
    })
    ...
});
```

In this test we describe the calculator. A special case of this test is that it(the calculator) should add two numbers. Now we call the calculate function with the operants 3 and 5 and the operator +. Because $3 + 5 = 8$ we also expect that the result equals 8.

If you run mocha now the test will fail, because there is no code which will be executed. Now we implemented the `calculate()` – function to fix this problem. When we are done with fixing this special problem, we are going to repeat this pattern. This means we write a new test and fix it with enhancements of the `calculate()` – function.

You can find the final code in our [Documentation-repository](#) and the results of the tests here:

```
Fabian@FABIAN-PC /C/Users/Fabian/test
$ mocha ./calculator.spec.js

calculator
  ✓ should add two numbers
  ✓ should subtract two numbers
  ✓ should multiply two numbers
  ✓ should divide two numbers
  ✓ should throw an error if you divide by zero
  ✓ should throw an error if you use an invalid operator
  1) should throw an error if you use an invalid operant

  6 passing (20ms)
  1 failing

1) calculator should throw an error if you use an invalid operant:
AssertionError: expected [Function: invalidOperant] to throw an error
  at Context.<anonymous> (<file>:43:37)
  at callFn (<file>:315:21)
  at Test.Runnable.run (<file>:308:7)
  at Runner.runTest (<file>:422:10)
  at c:<file>:53:12
  at next (<file>:342:14)
  at c:<file>:35:2:7
  at next (<file>:284:14)
  at Immediate._onImmediate (<file>:320:5)
```

```
Fabian@FABIAN-PC /C/Users/Fabian/test
$ mocha ./calculator.spec.js

calculator
  ✓ should add two numbers
  ✓ should subtract two numbers
  ✓ should multiply two numbers
  ✓ should divide two numbers
  ✓ should throw an error if you divide by zero
  ✓ should throw an error if you use an invalid operator
  ✓ should throw an error if you use an invalid operant

  7 passing (10ms)
```



15. April 2016 by CodeLionX on Allgemein • Edit→

Backend API Documentation Postman API

We have tested our API with the Google Chrome Plugin Postman. You can add our Collection of Tests to your Postman to see and run the tests as well. Use this [link](#).



14. April 2016 by CodeLionX on Allgemein • Edit→

HW13: FP calculation and time estimation (depricated)

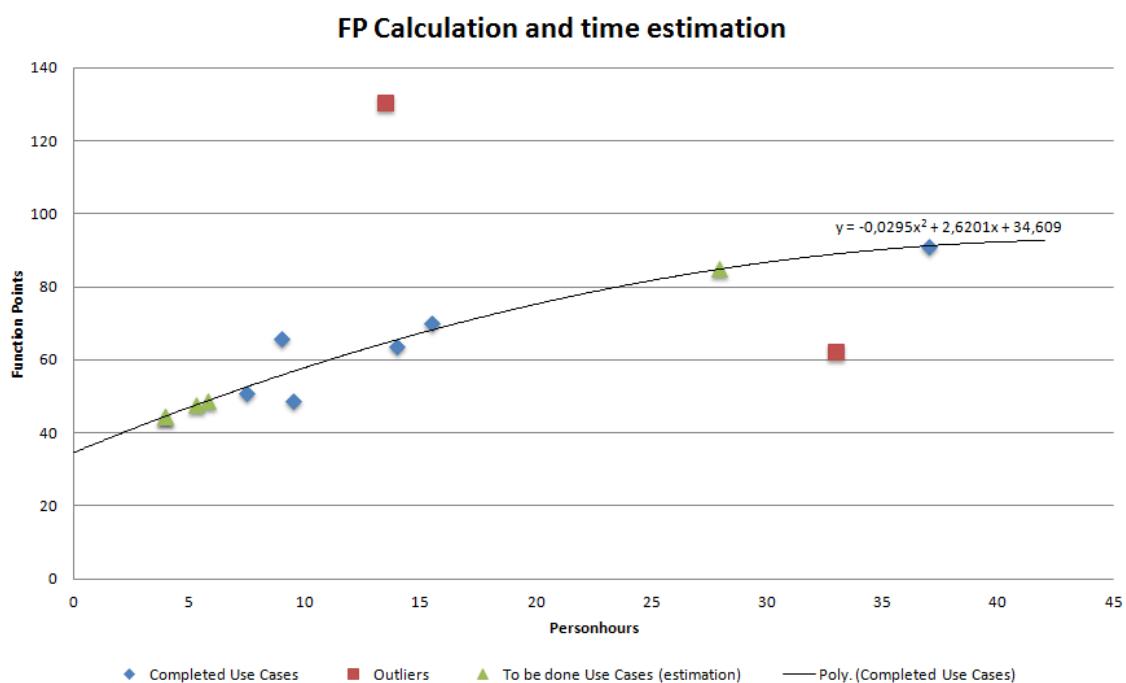
Hi folks!

The following table shows our already completed use cases with our time spent implementing it and the corresponding Function Points:

Use Case	Use Case Name	Total Time Spent (Estimation in h)	Function Points
(1	Capture and Stream Video	13,5	130,38)
2	Configure Settings	15,5	69,96
3	Maintain User Profile	9	65,72
4	Switch User	7,5	50,88
5	Register	14	63,60
6	Browse Media	33	62,40
7	Manage Users	9,5	48,76

We plotted this data in a graph with the Function Points on the ordinate and the total time spent on the abscissa. As you can see we have two outliers plotted in red. The first one at x=13,5 is the Use Case Capture and Stream Video. It is very complex and affects various files, therefore it has very much FPs. We have not completed this Use Case yet and have estimated a lower number of hours, because most of the time will go to the streaming library. Maybe this was a mistake and we have to correct our estimation, so that it matches with our chart.

The second outlier is the Use Case Browse Media, where we have spent too much time in spite of the low number of FPs. The reason for that was a change in our architecture and the technology we used for creating a web-based media browser. This lead thereto that we had to implement most of the functionality again.



We used this graph to estimate our remaining Use Cases.
You can see these estimations as green triangles in the graph.

You can find all our Use Case Specifications here:

[Documentation](#)

Have a nice evening,

your team Unveiled

EDIT:

There is a new version of the function point calculation and Use Case estimation. Please go to [this document](#).



Page 1 of 3

[Older Posts →](#)

Documentation links:

- [SRS](#)
- [SAS](#)
- [Use-Case: Capture and stream video](#)
- [Use-Case: Configure settings](#)
- [Use-Case: Maintain user](#)

profile



Casper WP by Lacy Morrow

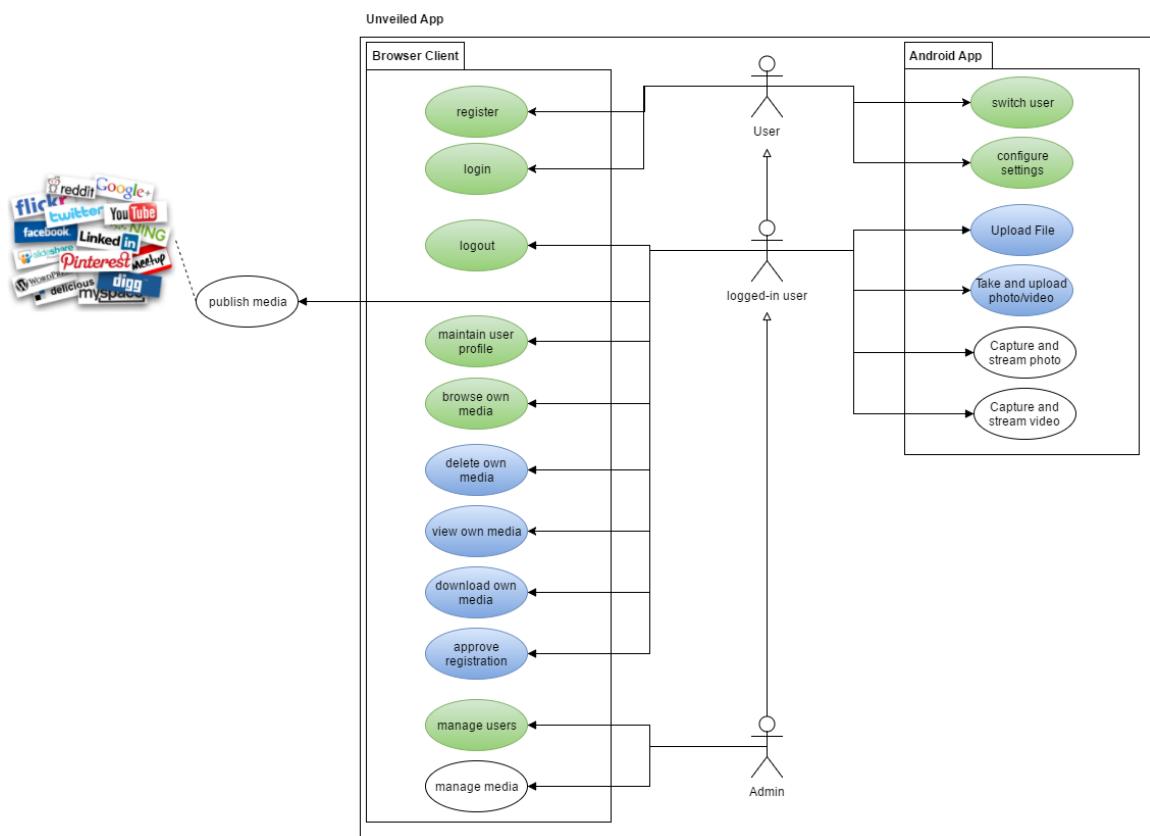


4. April 2016 by CodeLionX on Allgemein • Edit→

HW12: Scope for 2nd Semester and Risk Management

Hey there,

this is our new overall Use Case Diagram:



The green marked Use Cases were finished in the last semester and the blue ones show the scope for this semesters work on the Unveiled project. You can find all our Use Case specifications on our [Documentation site](#). These are the new ones (the blue marked Use Cases):

- [Use Case: Delete own Media](#)
- [Use Case: Download own Media](#)
- [Use Case: View own Media](#)
- [Use Case: Approve Registrations](#)
- [Use Case: Upload File](#)

Risk Management

We used our Google Drive folder to create a new dynamic spreadsheet which contains our list of possible risks. You can see the current status of the list below or in the [document](#):

Risk Rank	Risk Name	Risk Description	Risk Probability of Occurrence	Risk Impact	Risk Factor	Risk Mitigation	Person in Charge of Tracking
1	RTSP library won't finish	We are writing our own streaming library based on an existing	60%	100 %	60 %	Don't use streaming to transmit the data -> use standard upload	Fabian Schäfer

		approach. This is more complex than expected and may take longer to implement				d/download	
2	lib streaming doesn't work	The Android App will make use of an external library called "libstreaming"; when it does not fit into our architecture it will not work	50%	90 %	45 %	check dependencies and architecture early; use own library for streaming	Sebastian Adams
3	exam preparation disrupts project progress	in final project phase also exam preparation	100%	40 %	40 %	Make a detailed planning for this final project	Sebastian Schmidl

		takes place, therefore team members must prioritize and can not spend all their time for the project				t phase early enough; cut off requirements	
4	server contract expires	Server contract expires or 1&1 terminates the contract	80%	40 %	32 %	check contract periodically; search and prepare backup solution	Sebastian Adams
5	bad code quality	end product contains a lot of bugs or the performance of the application is poor	40%	60 %	24 %	Write tests; make code reviews	Fabian Schäfer
6	case	A	40%	30	12	divide	Sebasti

	of illness	team member sustains a little or serious injury or becomes sick for a longer time period		%	%	tasks of this person to the other team members; cut off some requirements	an Schmid l
7	server stops working	Private hosted server stops working, because of any issue	30%	30 %	9%	invest additional time on getting server running again; make data backups periodically; provide user guide with server set up steps	Sebastian Adams
8	team member leaves the university	Team member becomes exmatriculated or leaves	10%	70 %	7%	cut off requirements and use cases	Sebastian Schmid l

		the univer sity beaca use of perso nal reaso ns				
--	--	---	--	--	--	--

Use Case Estimation

The following table shows our time spent for the different Use Cases without all the work which would belong to multiple Use Cases:

Semester	Use Case	Use Case Name	Time spent (estimation in h)				Function Points
			Documentation	Coding	Testing	Total	
2	1	Capture and Stream Video	1,5	8	4	13,5	
1	2	Configure Settings	1,5	8	6	15,5	
1	3	Maintain User Profile	1	6	1	8	
1	4	Switch User	1	4	2,5	7,5	
1	5	Register	1	8	1	10	
1	6	Browse Media	1	34	2	37	

1	7	Manage Users	1,5	6	2	9,5	
---	---	-----------------	-----	---	---	-----	--



22. December 2015 by CodeLionX on Allgemein • Edit→

HW11: Midterm Presentation and Grading

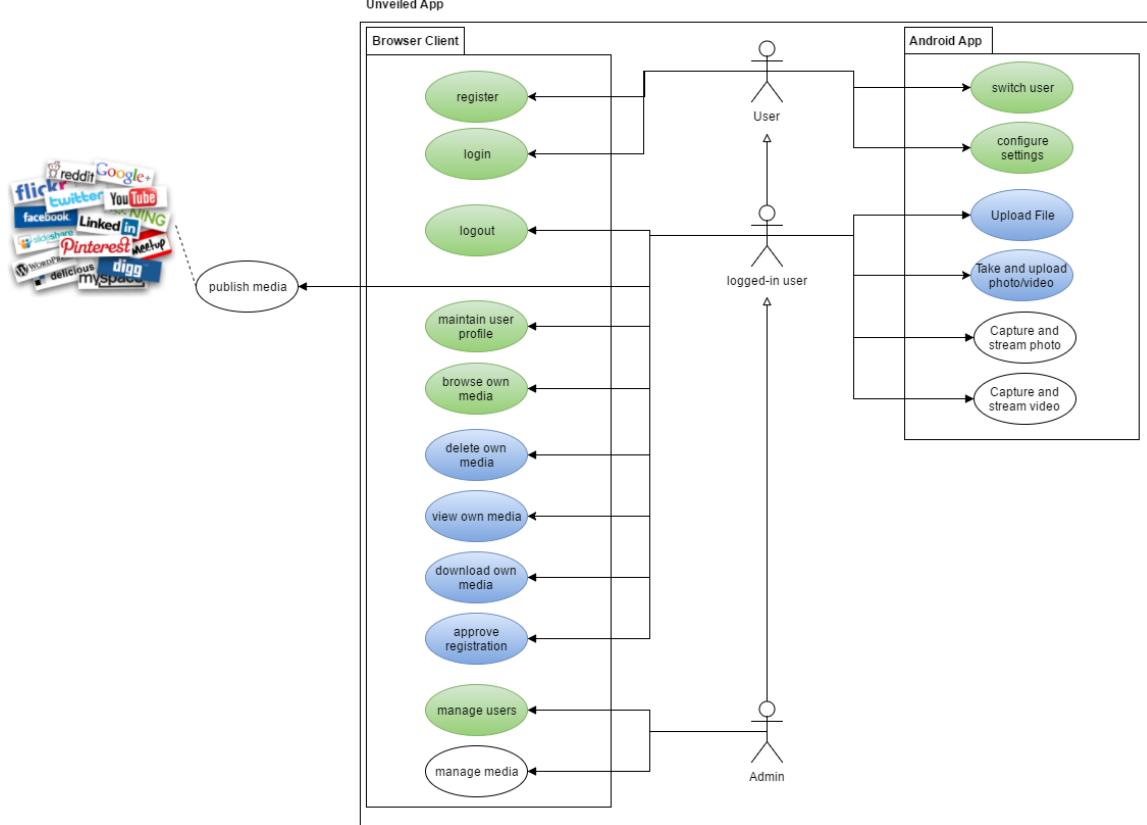
Hello together,

today we want to show you all the work we've done in the past semester. To get a quick overview you can scroll through our presentation uploaded on Github: [Unveiled MidtermPresentation V1.0.pptx](#).

We are team Unveiled and that's how we had split up the tasks among us:

Team		Unveiled Fight against injustice straightaway
Members	Roles	Time spent
Sebastian Adams	Deployment / Configuration Manager System Analyst Software Architect Backend Implementer	86h
Fabian Schäfer	Software Architect Frontend Business Process Analyst Designer Implementer	82h
Sebastian Schmidl	Deployment / Configuration Manager Project Manager Test Manager Implementer	117h

To get a better overview about our tasks, you can see all use cases in the next picture illustrated by a diagram. Green use cases are already done. As you can see, the main focus in this semester was on creating the web interface for managing both the content and the users. Most of the use cases are located in the web interface component because we want the Android app to be as lightweight and lean as possible.



You can find all documents including the SRS, use case descriptions and the SAD on our [Documentation-Page](#). Our test cases written in Gherkin are shown in the corresponding use case descriptions ([UC1: Capture and stream video](#), [UC2: Configure settings](#) and [UC4: Switch user](#)). The following picture shows the test log of the use case: [Configure settings](#):

CucumberTest: 7 total, 7 passed

1 m 1 s

[Collapse](#) | [Expand](#)

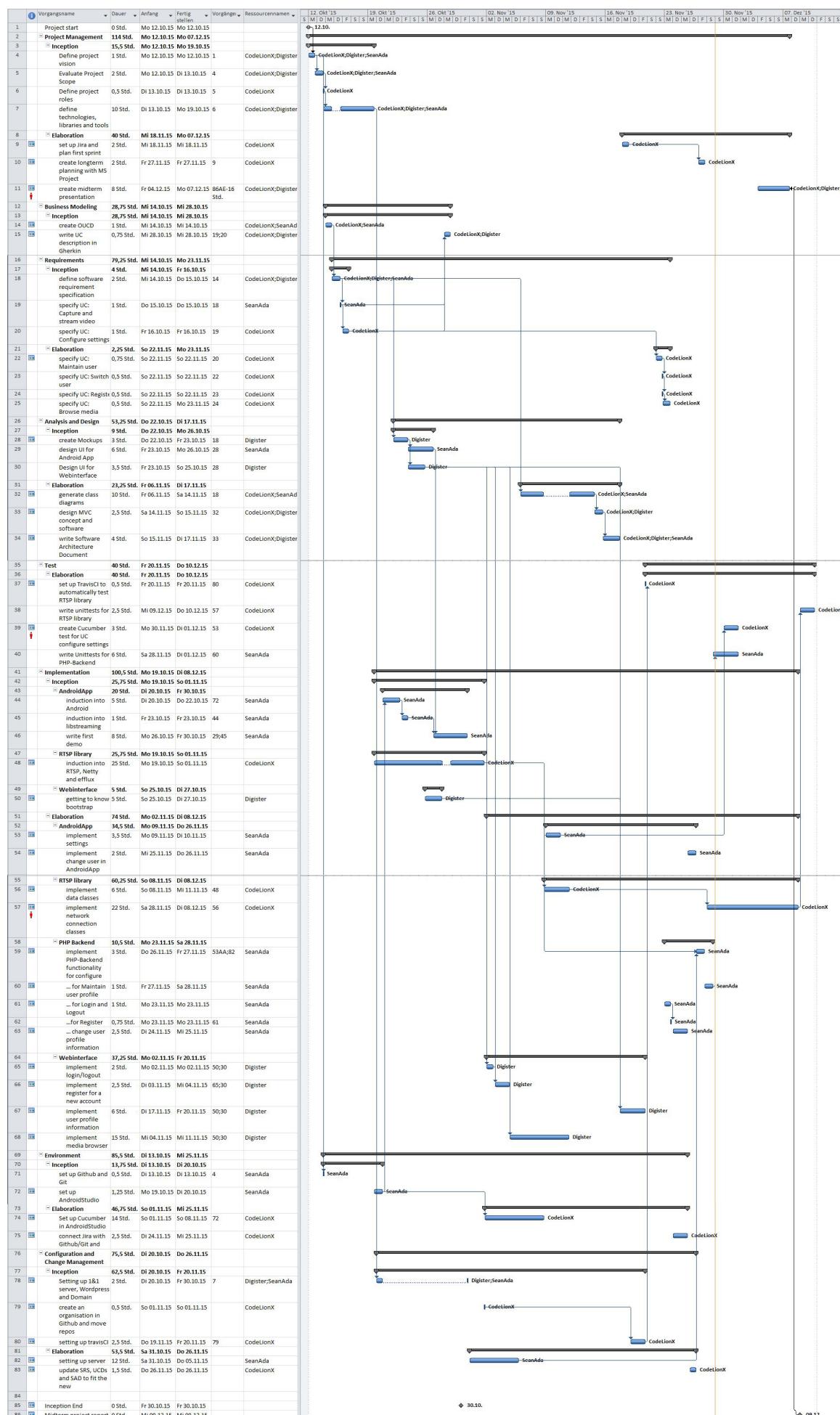
Feature Settings 1 m 1 s

Scenario Outline change connection settings	passed	2.27 s
Scenario Outline change connection settings	passed	6.00 s
Scenario Outline change connection settings	passed	6.85 s
Scenario Outline change connection settings	passed	8.15 s
Scenario Outline change video quality	passed	12.85 s
Scenario Outline change video quality	passed	11.54 s
Scenario Outline change video quality	passed	13.35 s

We have also done some unit testing for the server-side code and our own streaming library. You can find the test code in the related [Github-Repository \(SAS-Systems/imflux/test\)](#). These tests are also run by travisCI as you can see in the [travis-logs](#).

Regarding project management, you can find our Jira-Board and our Burndown-Diagrams [here](#). The long-term planning was made with MS Project and you can get the file from our [Github-Repository \(Unveiled.mpp\)](#). The following picture shows our complete Gantt-Chart.

@Ms. Berkling: We've discussed in class whether it's better to group the tasks by discipline or by phase first. We came to the decision that grouping the tasks by discipline first makes also sense.



Now we can show you the most important item: A **demo** of our application.

You can access our web interface [here](#). If you have an Android device, you can get our packed application from [Github \(Unveiled app.apk\)](#) and install it on your device or virtual device.

All our code is hosted by Github:

- AndroidApp: <https://github.com/SAS-Systems/Unveiled/tree/Android>
- Webinterface: <https://github.com/SAS-Systems/Unveiled/tree/Backend-PHP-Stack/Backend-PHP-Stack/webinterface>
- Backend PHP-Stack: <https://github.com/SAS-Systems/Unveiled/tree/Backend-PHP-Stack/Backend-PHP-Stack>
- Backend Java-Stack: not yet implemented (needs imflux to be finished first)
- imflux (streaming library): <https://github.com/SAS-Systems/imflux>

We wish you merry Christmas and a happy new year!!

Your team Unveiled

- CodeLionX
- Digister
- Seanada



29. November 2015 by CodeLionX on Allgemein • Edit→

HW9: GC_RUP

We have just finished our longterm-planning with MS-Project. You can find our Gantt-Chart at Github as [picture](#) or as [PDF](#). Please follow these links. We can't show you the picture here because it's quite too big.

We have also uploaded a picture of all our tasks assigned to the team members:

Ressourcenname	Nr	ch	12. Okt '15	S	M	D	M	D	F	S	S	19. Okt '15	S	M	D	M	D	F	S	S	26. Okt '15	S	M	D	M	D	F	S	S	02. Nov '15	S	M	D	M	D	F	S	S	09. Nov '15	S	M	D	M	D	F	S	S	16. Nov '15	S	M	D	M	D	F	S	S	23. Nov '15	S	M	D	M	D	F	S	S	30. Nov '15	S	M	D	M	D	F	S	S	07. Dez '15	D																	
CodeLionX			D	E	v	c	r	d	f	s	s	define	s	i	n	t	u	c	s	s	induction into RTSP, Netty and efflux	w	i	n	d	u	c	s	s	Set up Cucumber in AndroidStudio	i	m	l	e	g	o	u	u	implement dataclasses	g	u	u	u	u	u	u	u	generate sig Software	u	u	u	u	u	u	u	u	set up test	u	u	u	u	u	u	u	u	spoon	u	u	u	u	u	u	u	u	create cucumber project	u	u	u	u	u	u	u	u	write unitte	u	u	u	u	u	u	u	u	
Digister			D	E	v	c	r	d	f	s	s	define	s	i	n	t	u	c	s	s	Mock	u	i	l	o	o	u	u	u	u	Design setting to know	u	u	u	u	u	u	u	u	implement media browser	u	u	u	u	u	u	u	u	implement user profile	u	u	u	u	u	u	u	u	create cucumber	u	u	u	u	u	u	u	u	create midterm presentation	u	u	u	u	u	u	u	u																		
SeanAda			D	E	v	c	r	d	f	s	s	define	s	i	n	t	u	c	s	s	set Induction into technologies, UI into A	p	i	l	o	o	u	u	u	u	desc UI into first demo	u	u	u	u	u	u	u	u	setting up server	u	u	u	u	u	u	u	u	generate class-diagram	u	u	u	u	u	u	u	u	implement Software	u	u	u	u	u	u	u	u	change profile	u	u	u	u	u	u	u	u	surfice	u	u	u	u	u	u	u	u	create midterm presentation	u	u	u	u	u	u	u	u

If you want to open our Gantt-chart with MS Project, just use this link: [Unveiled Gantt Midterm.mpp](#)

In addition we have done the descriptions and diagrams of a couple of new use cases. We will add Gherkin descriptions and more screenshots soon. You can find all our documents on the following page: <http://unveiled.systemgrid.de/wp/docu/>

Have a nice day
– team Unveiled



20. November 2015 by CodeLionX on Allgemein • Edit→

HW8: Setup Jira Scrum Board

Today we want to show you, how we are managing our agile software development process. This is done with a Jira-Board which you can access under this URL: <http://jira.it.dh-karlsruhe.de:8080/secure/RapidBoard.jspa?rapidView=10&projectKey=UNV>.

We have started our first sprint today and are already working on the tasks. The RUP workflows are assigned to every task through keywords. But most of them are also named like the corresponding workflow (e.g. *implement login page*). Our stories have subtasks, which represent the different small tasks that have to be done to complete the story. We have assigned the workload to every story as well, but unfortunately our subtasks don't have an estimated time... In the next sprint we will correct this little fault for sure.

So far it wasn't possible for us to connect our Github- and Git-Accounts to Jira, because the needed settings option isn't enabled yet.

We hope you like our blog and wish you all the best.

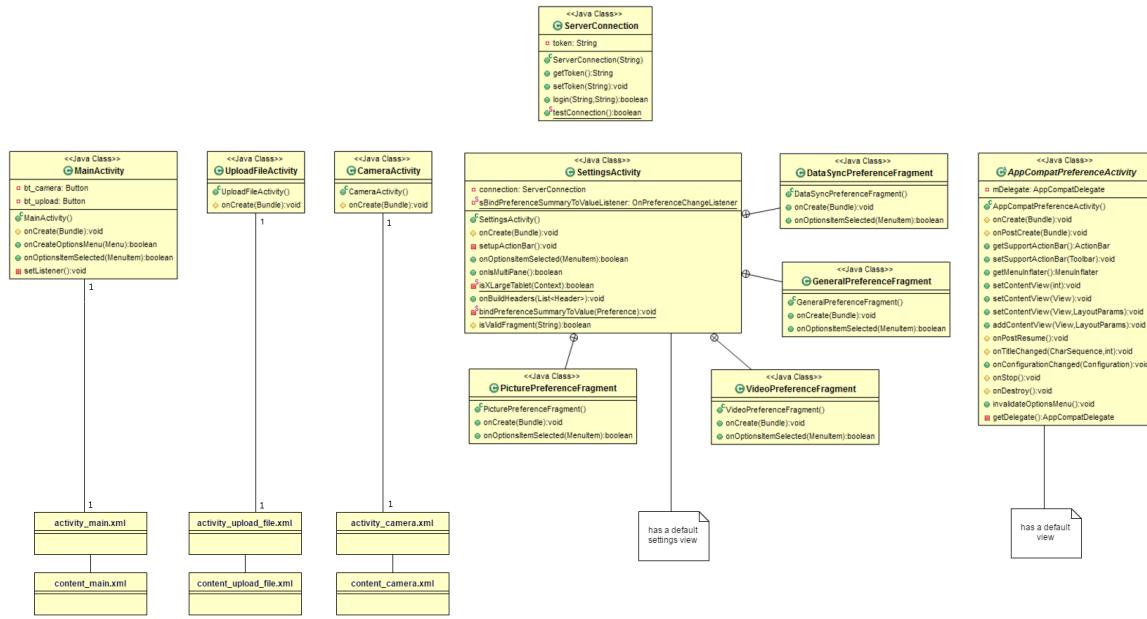


15. November 2015 by Digister on Allgemein • Edit→

HW7: MVC

This week we show you our MVC concept and also a demo version of our first use case. You can get detailed information of our concept and used framework in our [SAD](#). Because of the fact that our application is done for streaming we haven't any CRUD use case. To access the demo version of our application click the following [link](#).

Also we have automatically generated an UML diagram which you can see below. It is the class diagram of our Android App containing components from the controller and view part of the MVC concept.



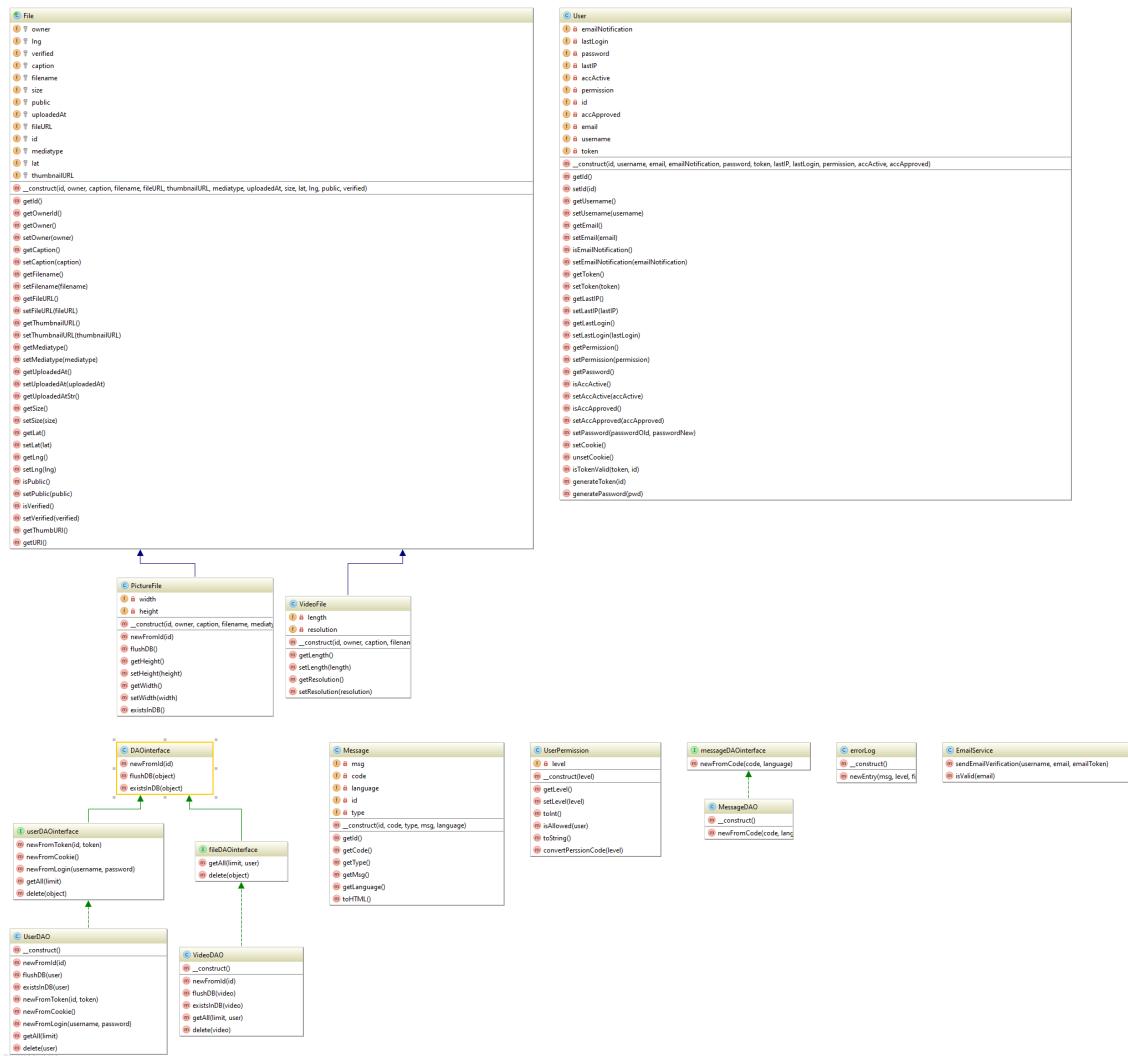
If you have any questions regarding this blog post or our project feel free to ask.

8. November 2015 by CodeLionX on Allgemein • Edit→

HW6: Class Diagram

Today we want to show you our first class diagrams. We decided to make our first class diagram with JetBrains PhpStorm. The following UML diagram shows our server-side API for our soon-to-be website, on which you will be able to explore your uploaded content and manage your profile.

This class diagram is work in progress, because we will continue developing more and more classes. Thus it's possible that some classes, method signatures and relations can change. In this case we will let you know and update our diagrams.



If you have any questions regarding this blog post or our project feel free to ask.

1. November 2015 by Digister on Allgemein • Edit→

HW5: Use cases transformed into features

This week we transformed our two use cases from HW4

into two features written in Gherkin. Each feature is separated in scenarios. A scenario is an order of steps within the use case describing one way through the use case diagram, for example “change server settings” or “change video quality settings”. Each of the following pictures shows a feature separated in different scenarios. For a better understanding every feature file is combined with its use case diagram in the following pictures.

For our tests, we will use the Cucumber-Plugin for Android Studio. The following picture shows the syntax highlighting and the code completion function of our IDE:



Feature one: Capture and stream video

```

Feature: capture video and stream it
  As a logged-in user
  I want to capture a video with my Android smartphone
  So that I can see a preview
  and the video is streamed to the server

  # straight forward scenario:
  @wip
  Scenario: start video preview
    Given all settings are set
    And I see homescreen
    When I press the button "capture video/photo"
    Then I should see a video preview on the display

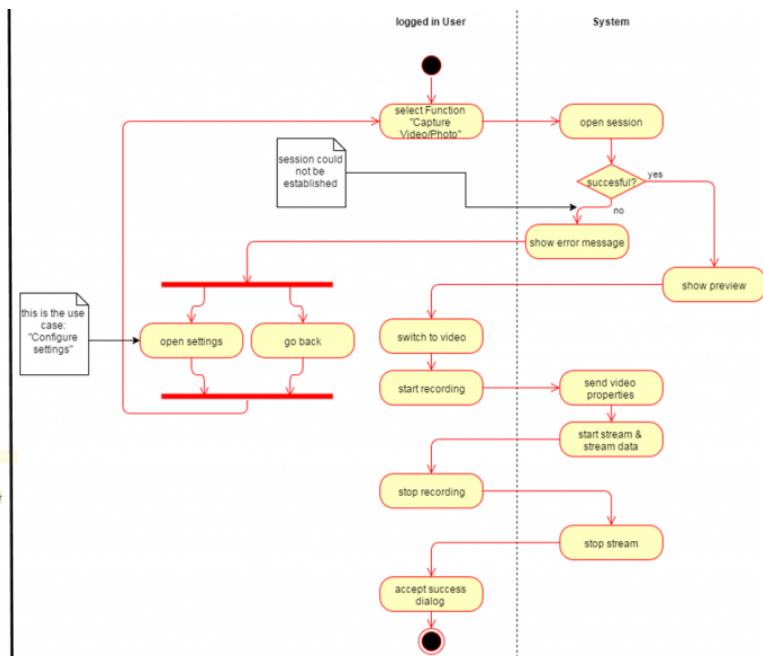
  @wip
  Scenario: start recording
    Given I see a video preview
    When I press the button "start recording"
    Then I should still see a video preview on the display
    And I should see a sending indicator on the display

  @wip
  Scenario: stop recording
    Given I see a video preview
    And I see a sending indicator
    When I press the button "stop recording"
    Then I should see "video successfully streamed" on the display
    And I should be redirected to "homescreen"

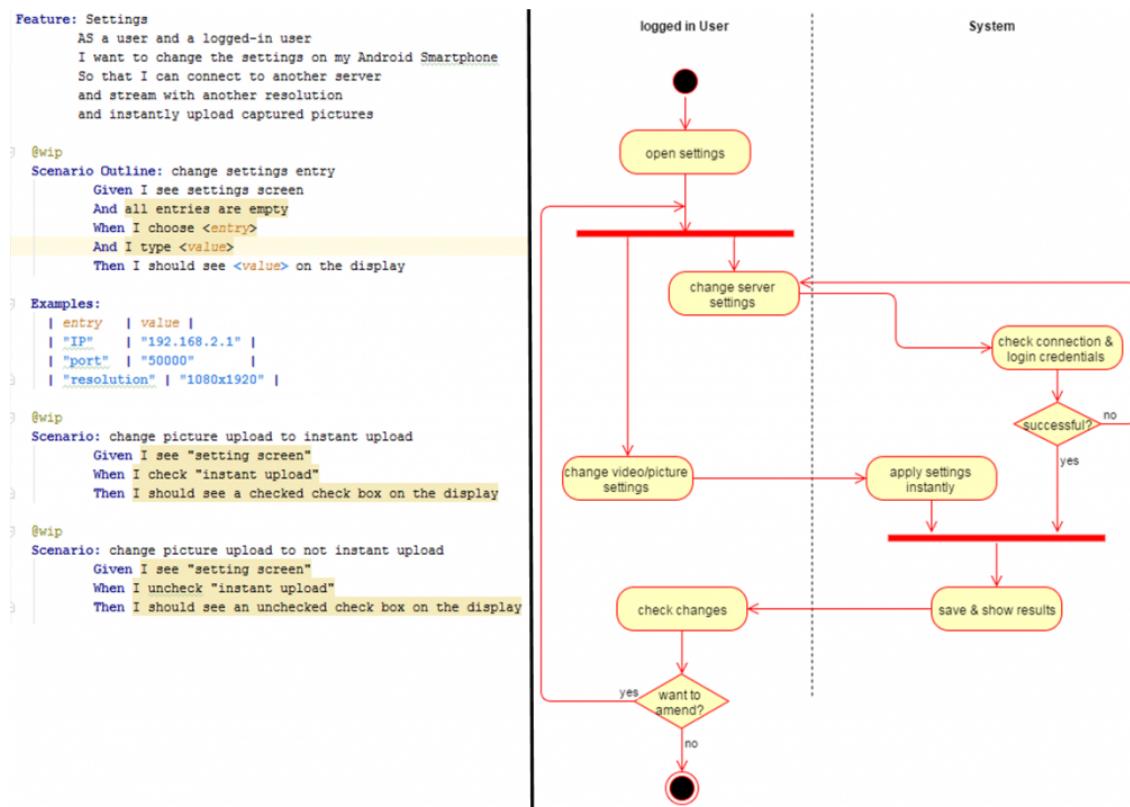
  # can't connect to the server
  @wip
  Scenario: failed to login
    Given all settings are set
    And I see homescreen
    And The server is not available
    When I press the button "capture video/photo"
    Then I should see "Failed to connect to server!" on the display

  @wip
  Scenario Outline: go back after failed login
    Given I see "Failed to connect to server!"
    When I press the button <screen>
    Then I should see <screen> on the display

  Examples:
    | <screen> | <screen> |
    | "back"   | "homescreen" |
    | "open settings" | "settingscreen" |
  
```



Feature two: Configure settings



25. October 2015 by Digister on Allgemein • Edit→

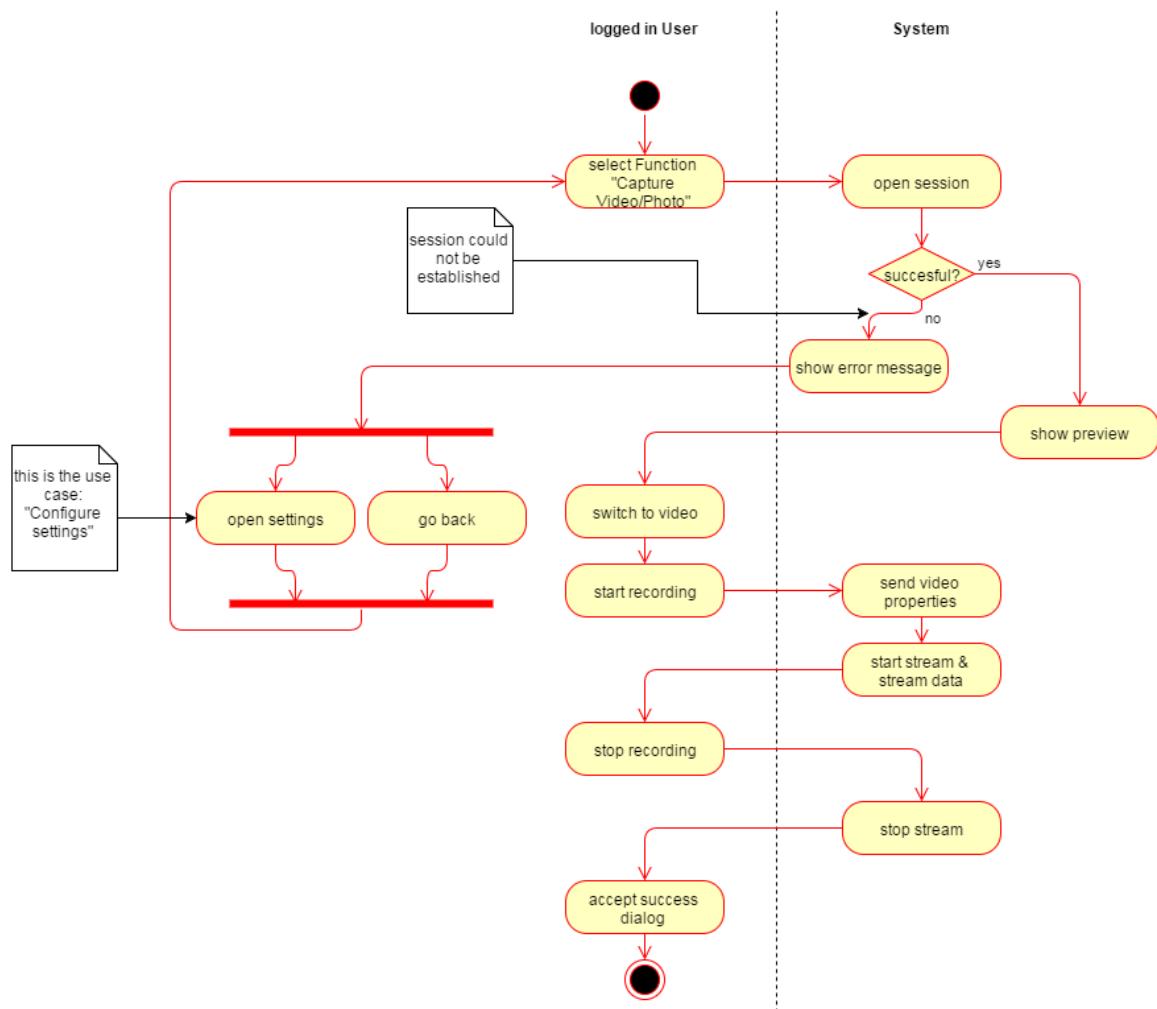
HW4: Detailed use cases

This week we described two use cases from our overall use case diagram, which you can view [here](#). The detail analyzed use cases are “capture and stream video” and “configure settings”. It is clearly understandable that points like special requirements can’t be determined yet. But we will keep you up to date if something alters. Also we changed our

versioning from Google Docs to Github, which you can access [here](#).

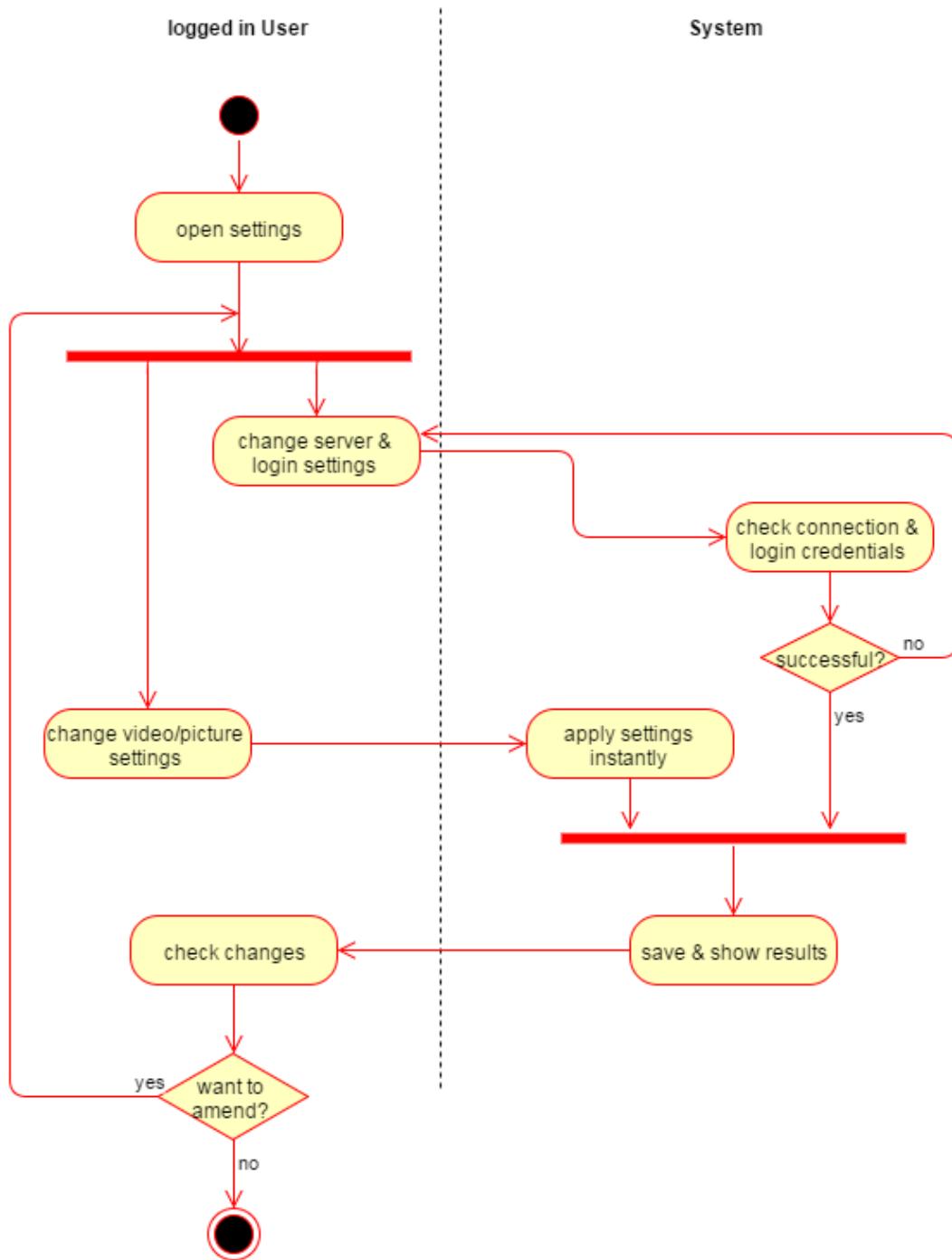
Use Case: Capture and stream video

This diagramm shows the basic flow of capture and stream video.



Use Case: Configure settings

This diagramm shows the basic flow of configure settings.



HW3: SRS and UCD

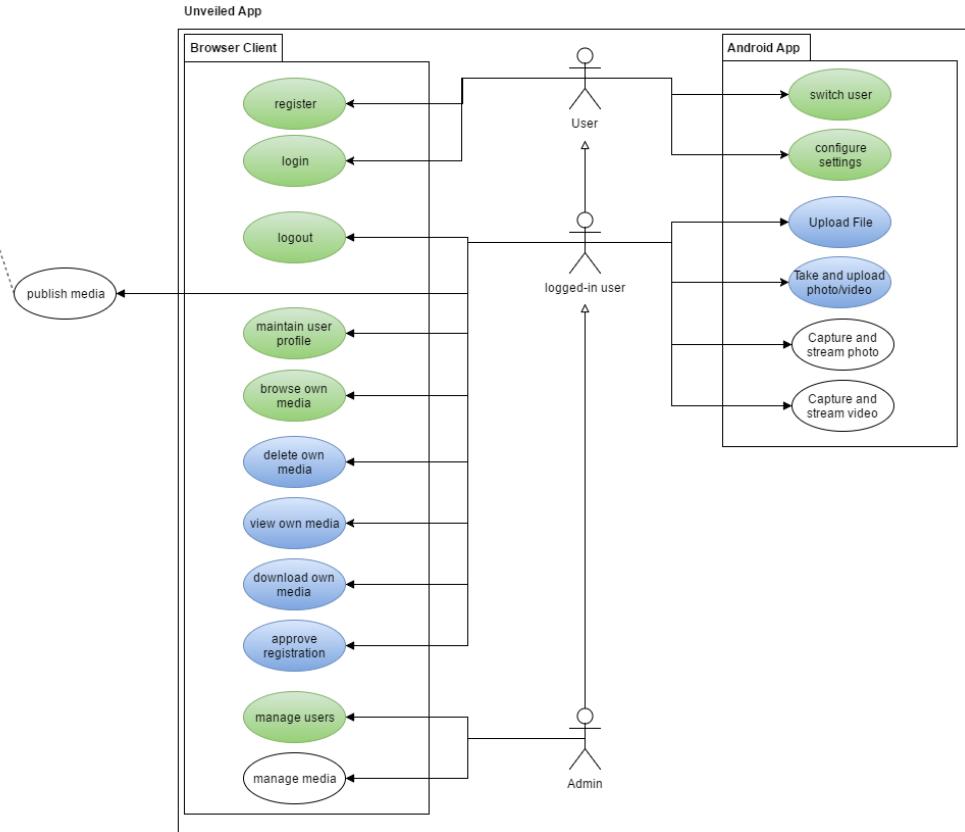
Software Requirements Specification (SRS)

Today we want to show you, which requirements our software should fulfill. We use GoogleDocs and GoogleDrive to version control our documents. You can access our current state of the SRS [here\(GoogleDrive\)](#) [Edit: [here\(Github\)](#)] and the public version [here\(Github\)](#) [Edit: [here\(Website\)](#)]. In GoogleDrive we have kept some blue comments to easily find chapters which aren't finished yet. Use Cases, User Interface Mockups and further specifications are following soon.

We are currently discussing about moving all documentation files to Github and manage them with Git. We let you know, if something changes.

Overall Use Case Diagram (OUCD)

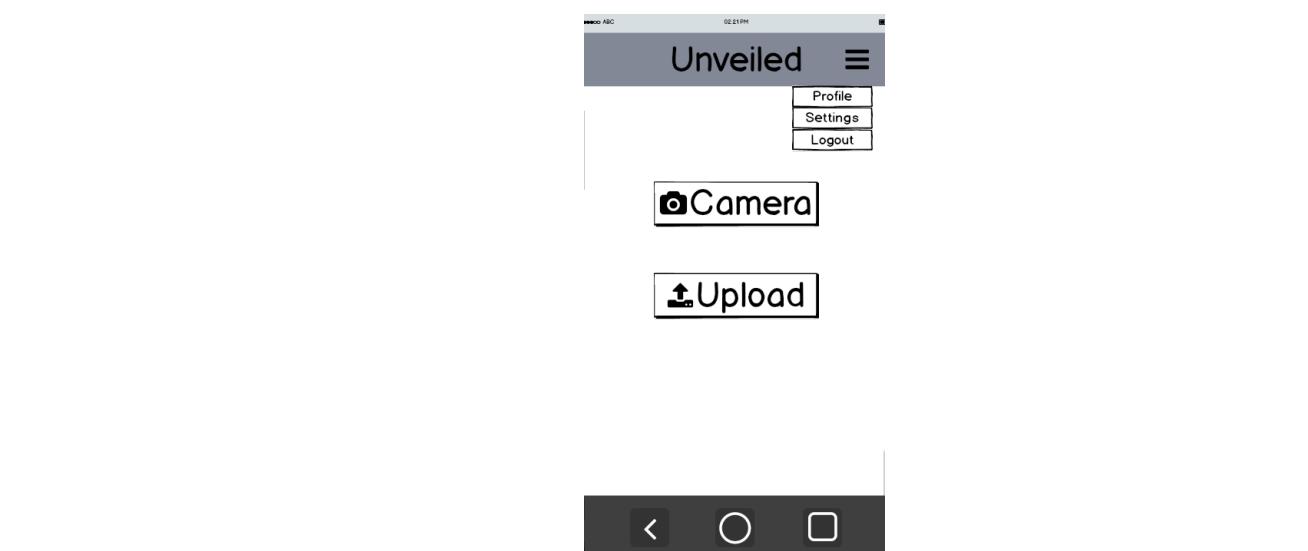
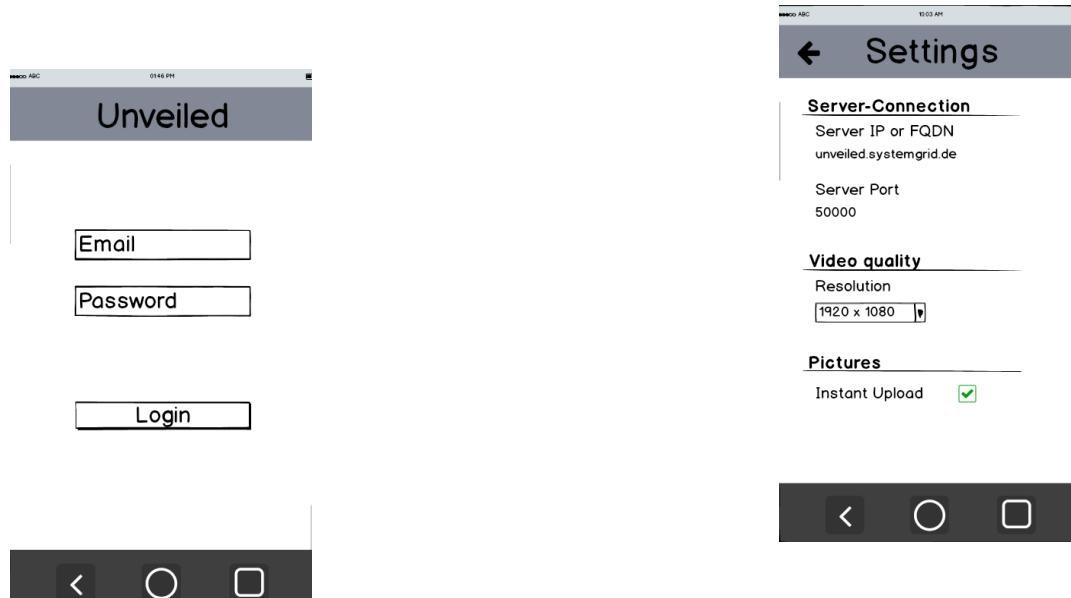
The following picture shows the latest version of our OUCD:



Also check out our documentation repository on Github at
[Unveiled Documentation](#)

Edit: Screenshots and GUI-Mockup for Android App

We've just finished our first mobile App Mockup for our Unveiled Android App. In this design we used paper-prototyping in combination with [POP](#). With POP you can create a “working” prototype out of pen and paper ideas. You can find some screenshots below. All screenshots of our first design are on [Github](#).



Edit 2: Changed management for documentation

Documentation files are now written in `markdown` and managed with Git and Github. You can find all public documents [here](#). Our SRS can now be found on this site: <http://unveiled.systemgrid.de/wp/docu/docusr/>.

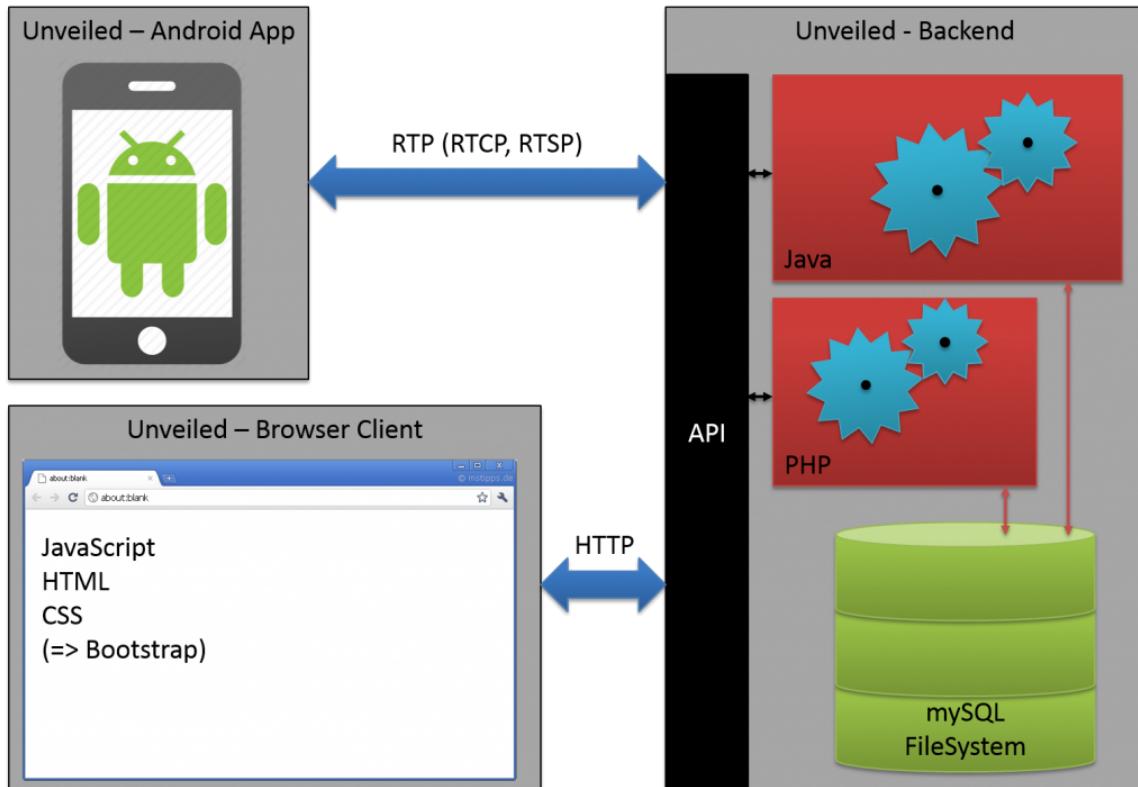


5. October 2015 by Digister on Allgemein • Edit→

HW2: Architecture and team roles

Architecture of our application and used technologies

We want to build an Android-Application that is able to instantly stream captured video material to our API. Therefore we will use RTP to stream the data via UDP. To control this data stream RTCP and RTSP are used. The receiving part of our backend is implemented in Java and supports these streaming protocols. The second part of our backend is implemented in PHP and provides a simple API for a browser client. This client provides users the possibility to browse, edit and publish their own media uploaded with our Android-App.



RUP team roles

	Role	Description
Sebastian Adams	Deployment / Configuration Manager	Oversees deployment and sets up development environment
	System Analyst	Discovers all requirement use cases
Fabian Schäfer	Software Architect Backend	Decides on technologies and patterns for the backend
	Software Architect Frontend	Decides on technologies and patterns for the frontend
Sebastian Schmidl	Business Process Analyst	Discovers all business use cases
	Designer	Details the analysis and design for use cases
	Test Manager	Select tests and ensures that testing is complete
	Project Manager	Plan, tracks and manages project
	Implementer	Develop and implement specific functionalities

[← Newer Posts](#)

Page 2 of 3

[Older Posts →](#)

Documentation links:

- [SRS](#)
- [SAS](#)
- [Use-Case: Capture and stream video](#)
- [Use-Case: Configure settings](#)
- [Use-Case: Maintain user profile](#)
- [Use-Case: Switch user](#)



[Casper WP](#) by Lacy Morrow



28. September 2015 by Digister on Allgemein • [Edit→](#)

Our Vision

Everyday a huge number of undetected crimes are committed on this planet. Whistleblowers try to impart those crimes to the community although politics and public authorities put rocks in their way. This project shall help these journalists and dedicated individuals to publish and save their captured video and photo material wherefore they have perhaps put their life at risk. Our Application addresses exactly this point.

Citizens with a conscience are not going to ignore wrong-doing simply because they'll be destroyed for it: the conscience forbids it.

Edward Snowden

People using our App can take pictures and videos which are immediately uploaded to our servers. There they are stored in a private library only accessible to the owner. Through a web based interface you are able to publish your recorded content.

28. September 2015 by Digister on Allgemein • Edit→

Our Blog

This is the blog written by the developers of Unveiled. Our team consists of three guys named Schmidl, Adams and Schaefer.

This blog deals with software engineering principles and methods which are needed for a successful project. Every progress and change will be tracked in this blog to create a comprehensible project schedule.

[← Newer Posts](#)

Page 3 of 3

Documentation links:

- [SRS](#)
- [SAS](#)
- [Use-Case: Capture and stream](#)
[video](#)
- [Use-Case: Configure settings](#)
- [Use-Case: Maintain user](#)
[profile](#)