

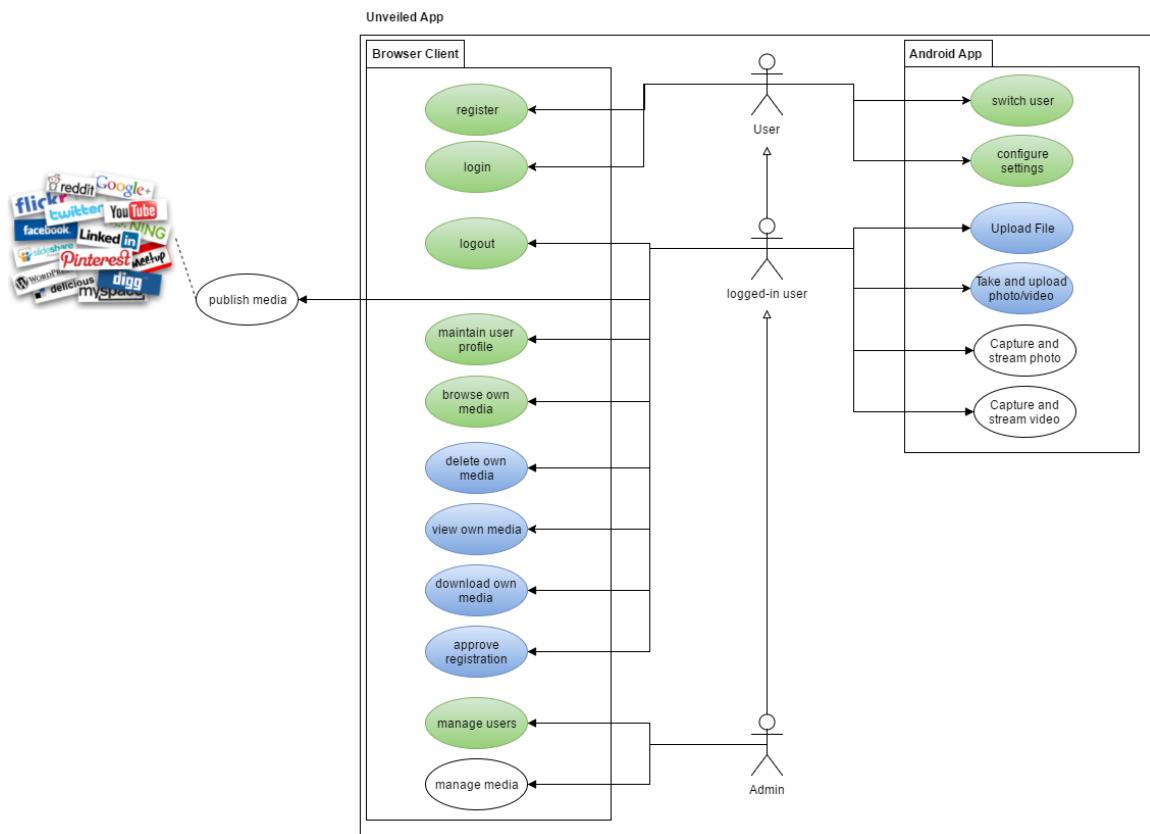


4. April 2016 by CodeLionX on Allgemein • Edit→

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

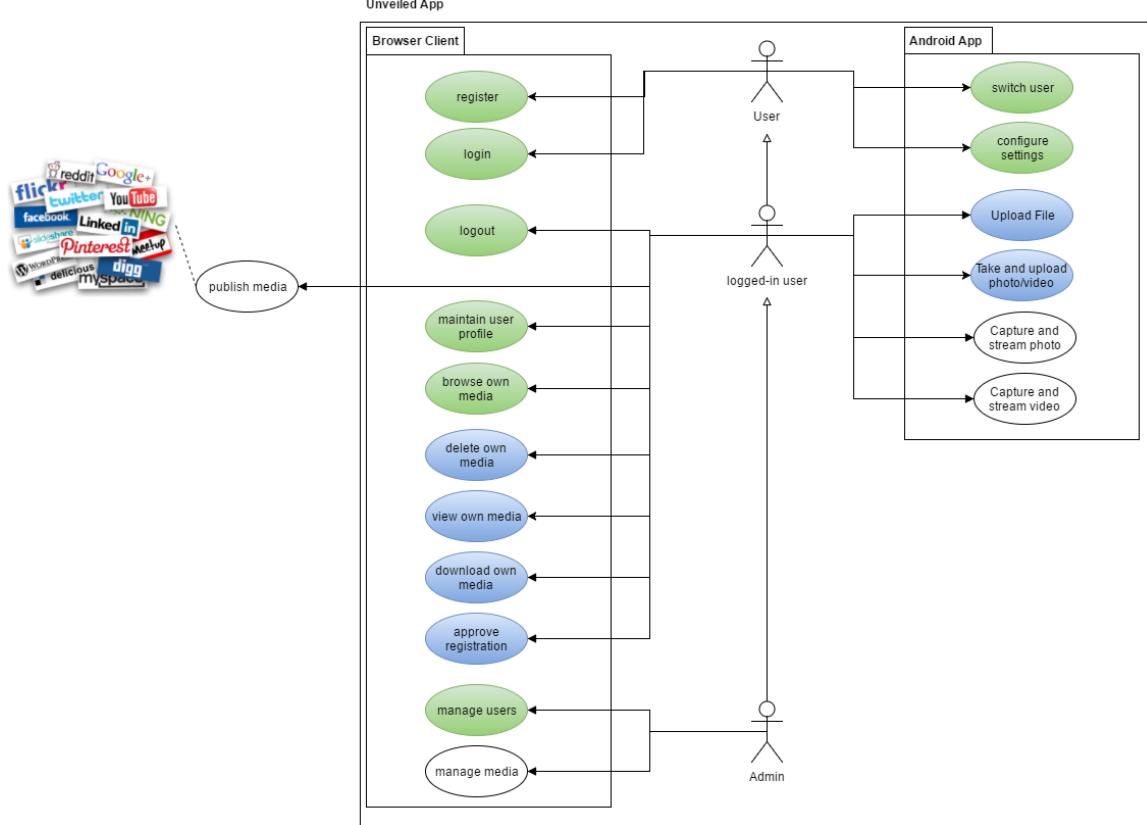
Hello together,

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](#)

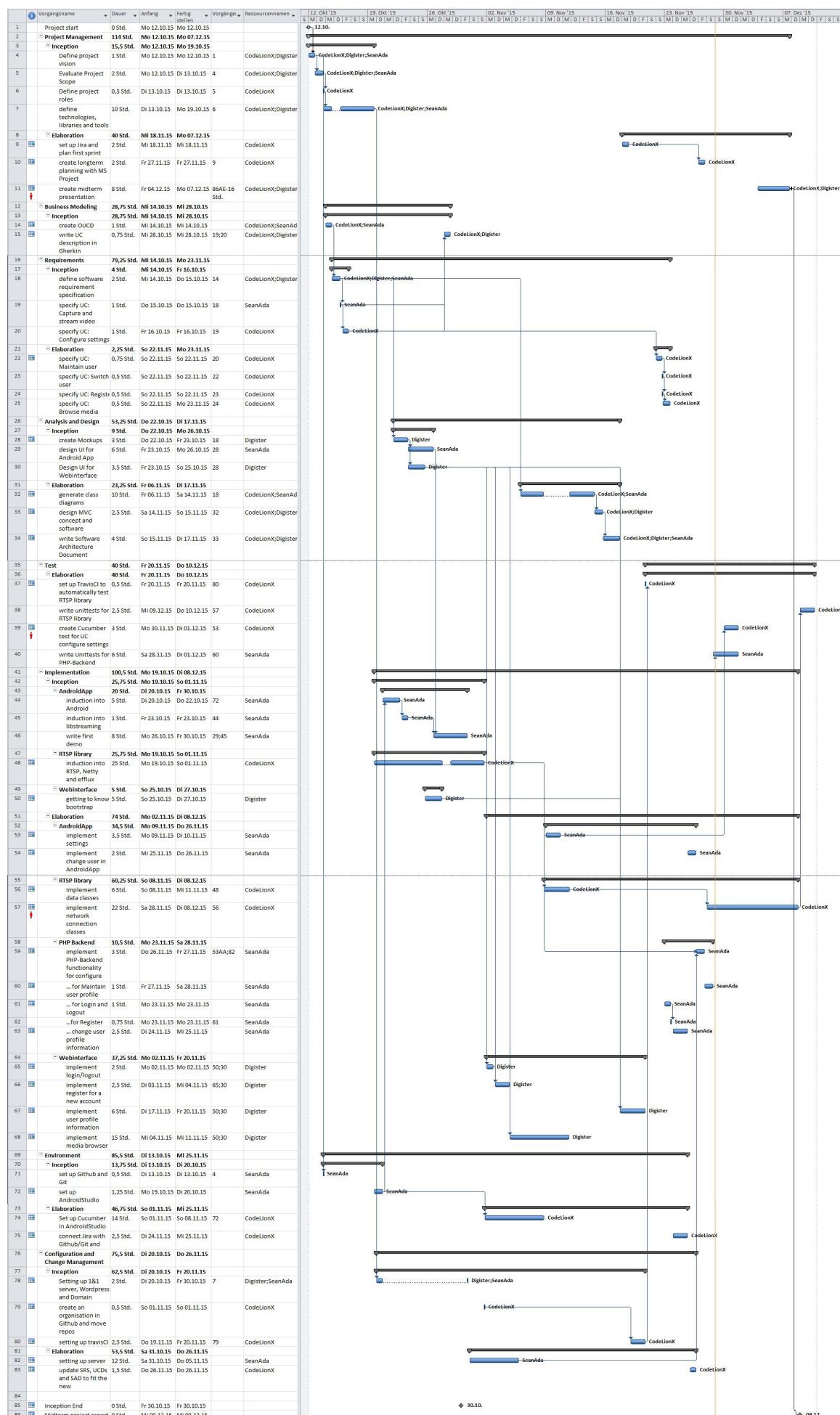
Feature Settings 1 m 1 s

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.

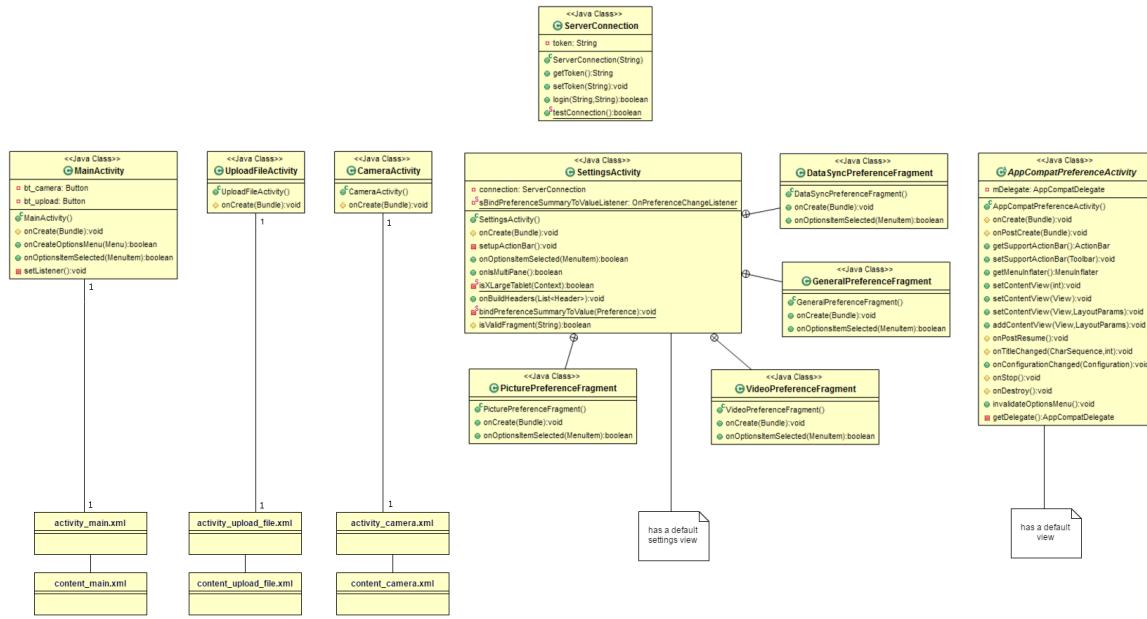


15. November 2015 by Digister on Allgemein • Edit→

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.



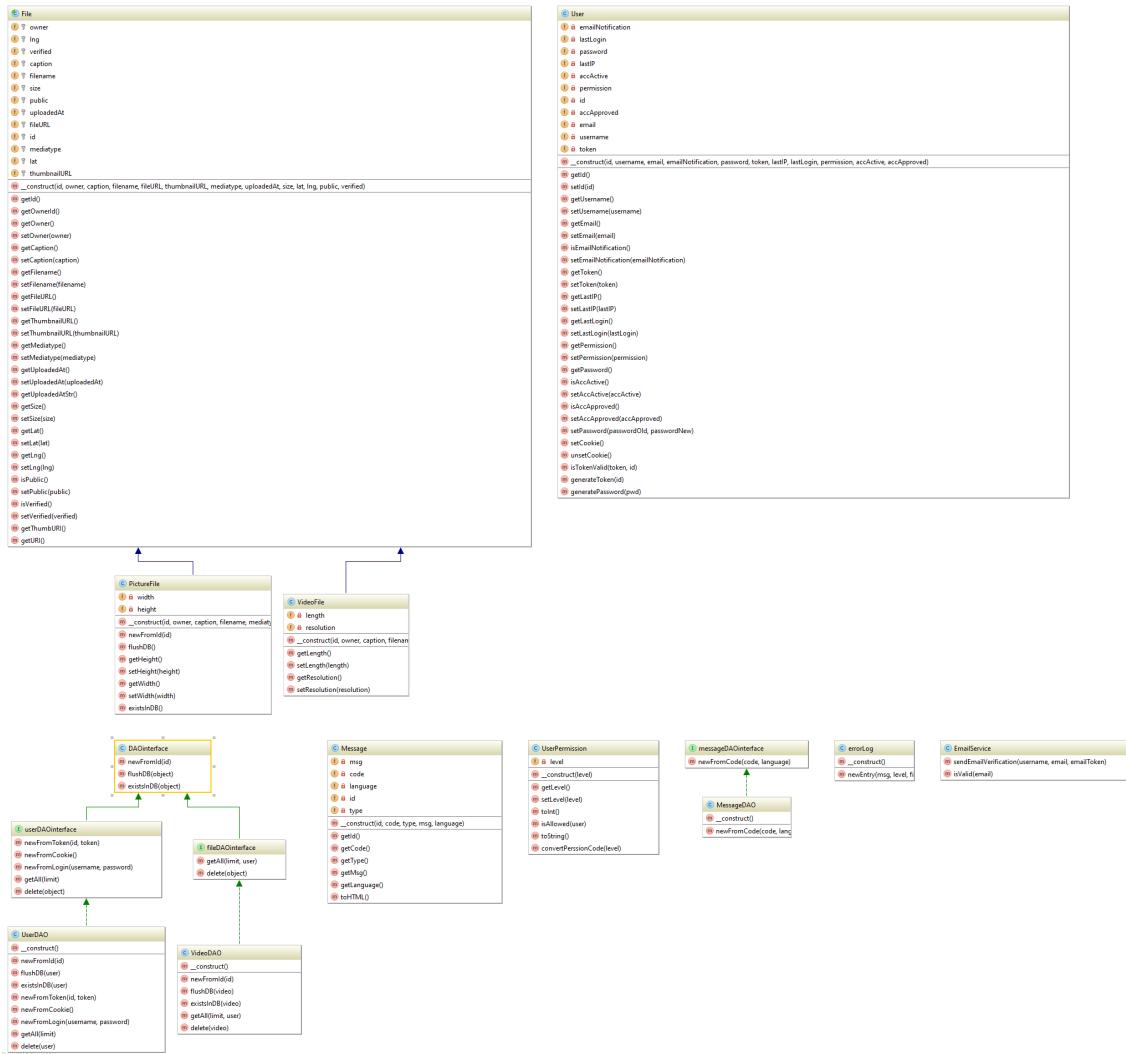
If you have any questions regarding this blog post or our project feel free to ask.

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

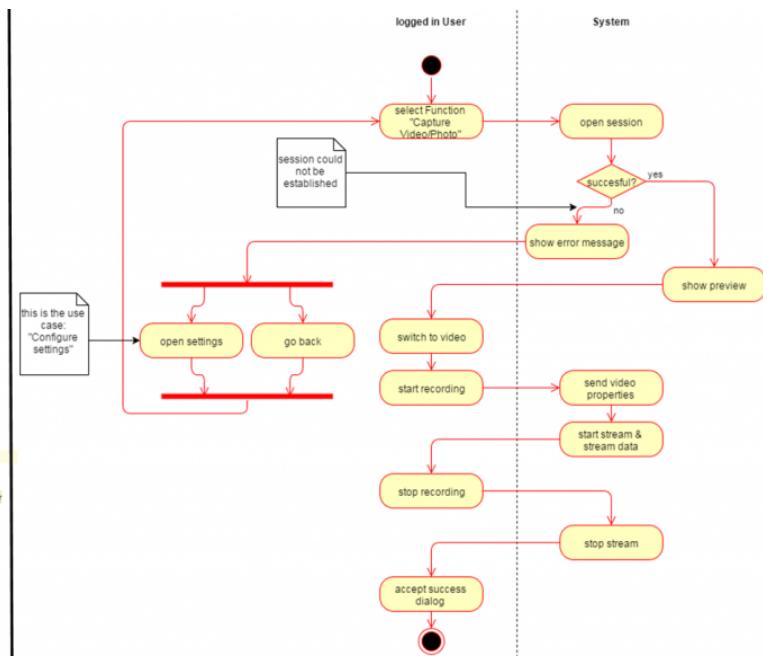
  @wip
  Scenario: start recording
    Given I see a video preview
    When I press the button "start recording"
    Then I should still see a video preview on the display
    And I should see a sending indicator on the display

  @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"

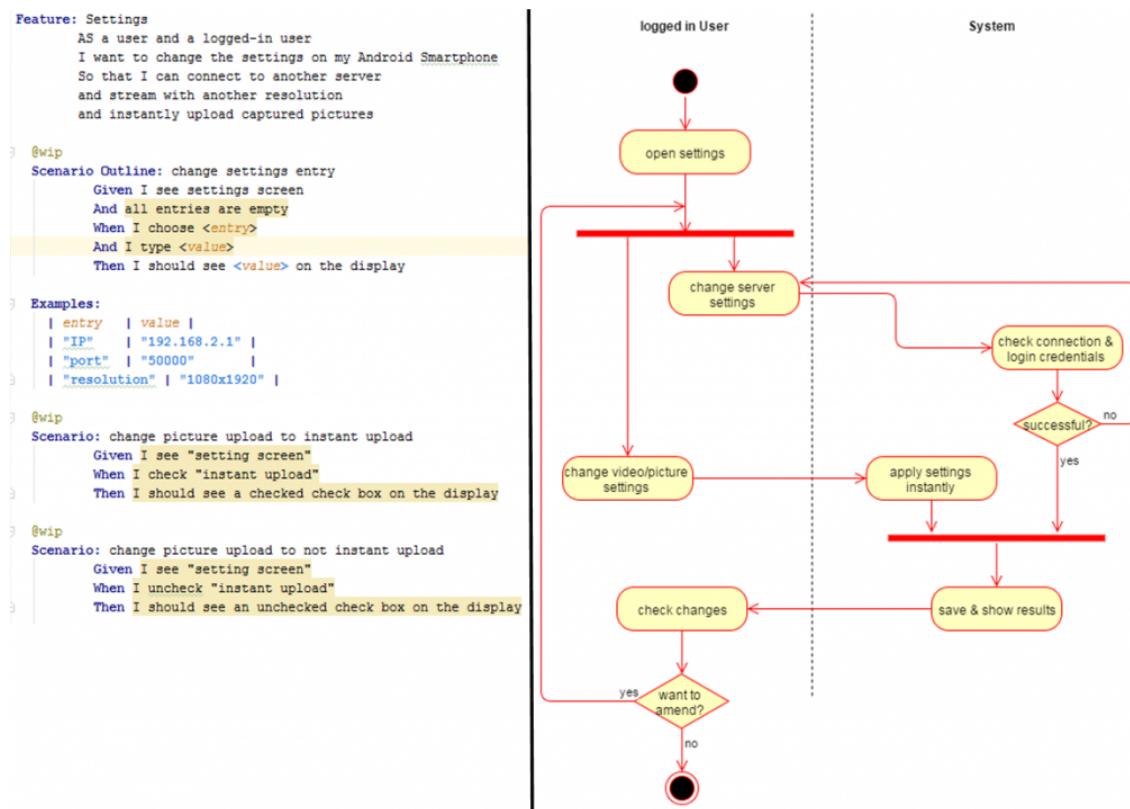
  # can't connect to the server
  @wip
  Scenario: failed to login
    Given all settings are set
    And I see homescreen
    And The server is not available
    When I press the button "capture video/photo"
    Then I should see "Failed to connect to server!" on the display

  @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" | "settingsscreen" |
  
```



Feature two: Configure settings



25. October 2015 by Digister on Allgemein • Edit→

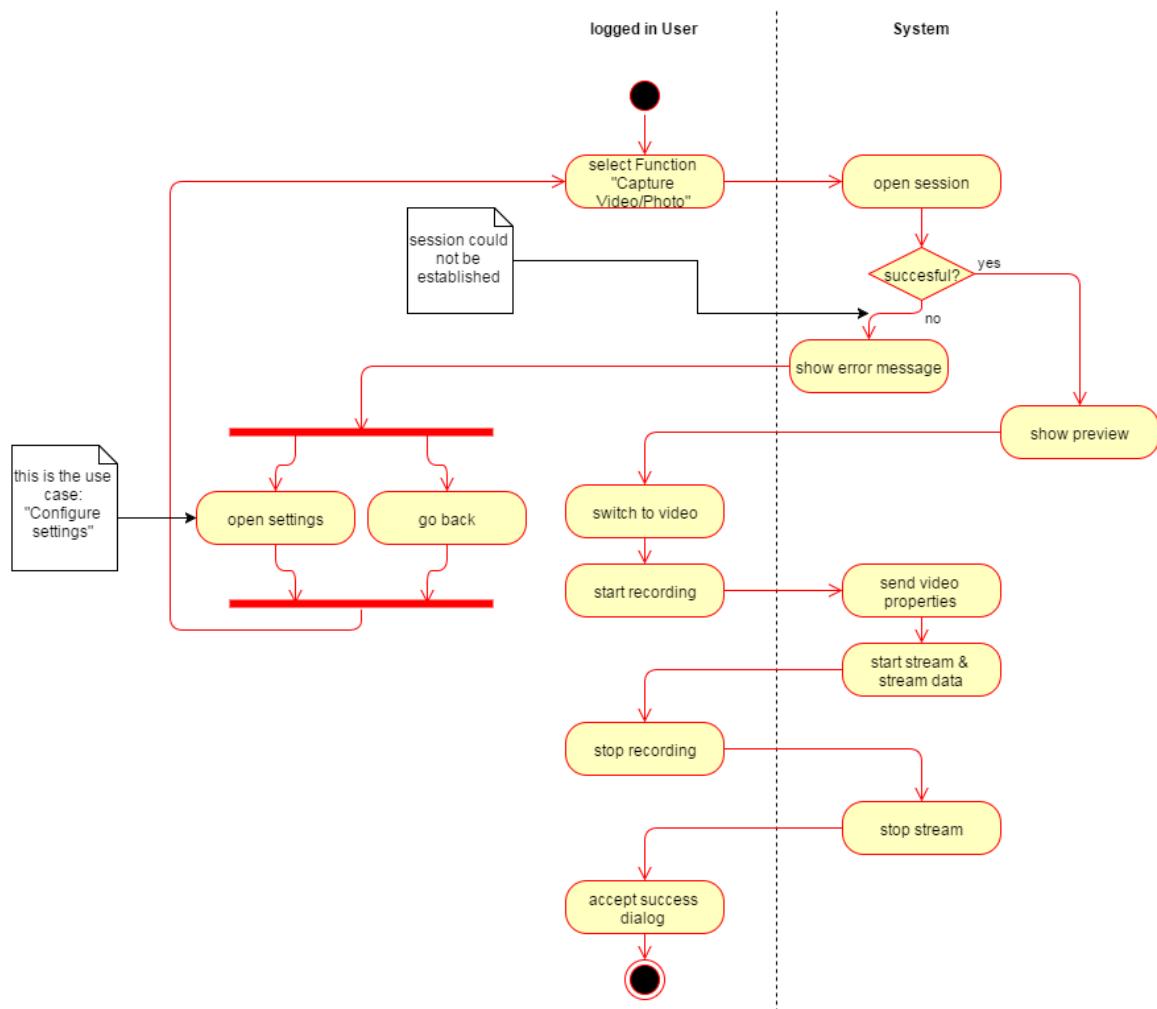
HW4: Detailed use cases

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](#).

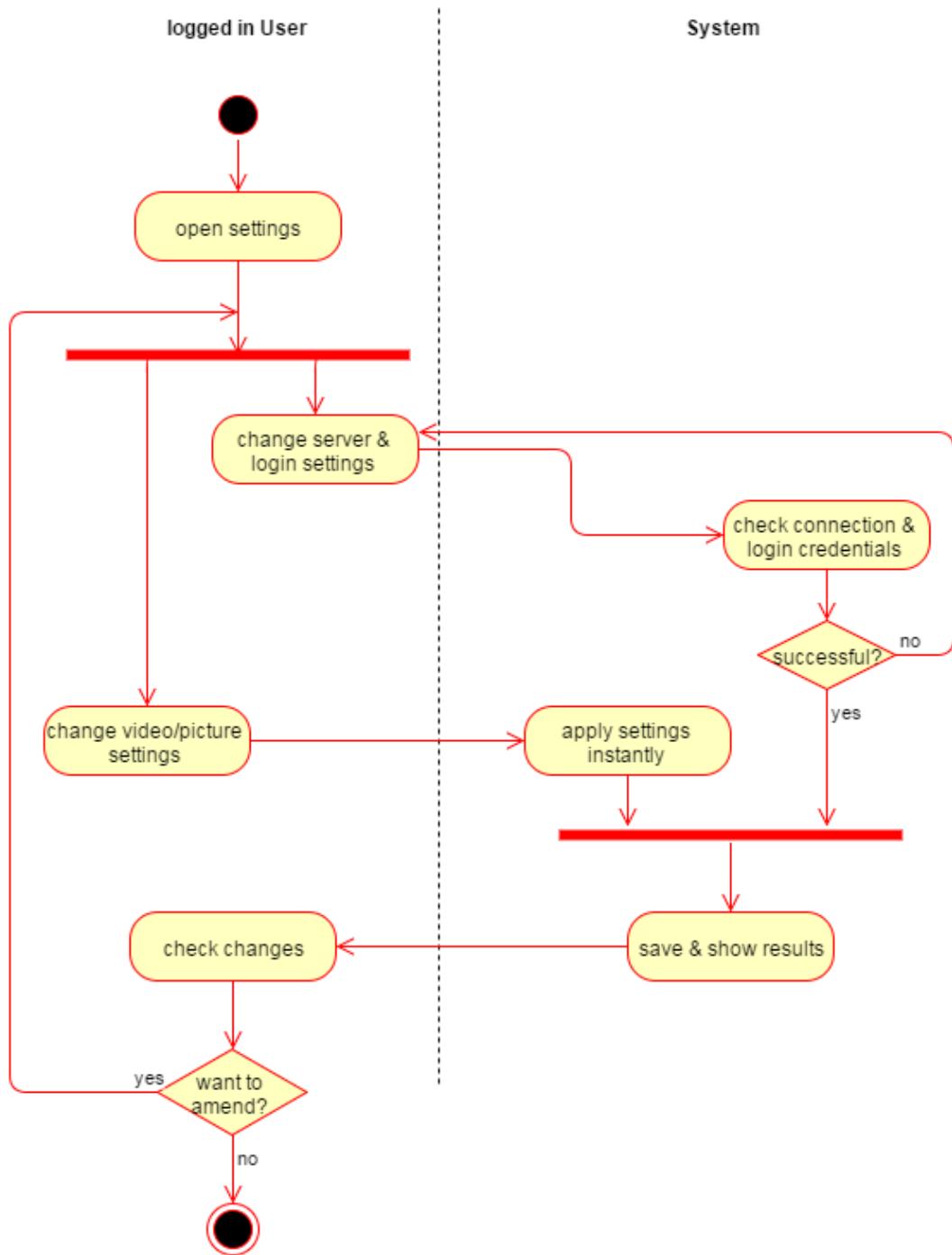
Use Case: Capture and stream video

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

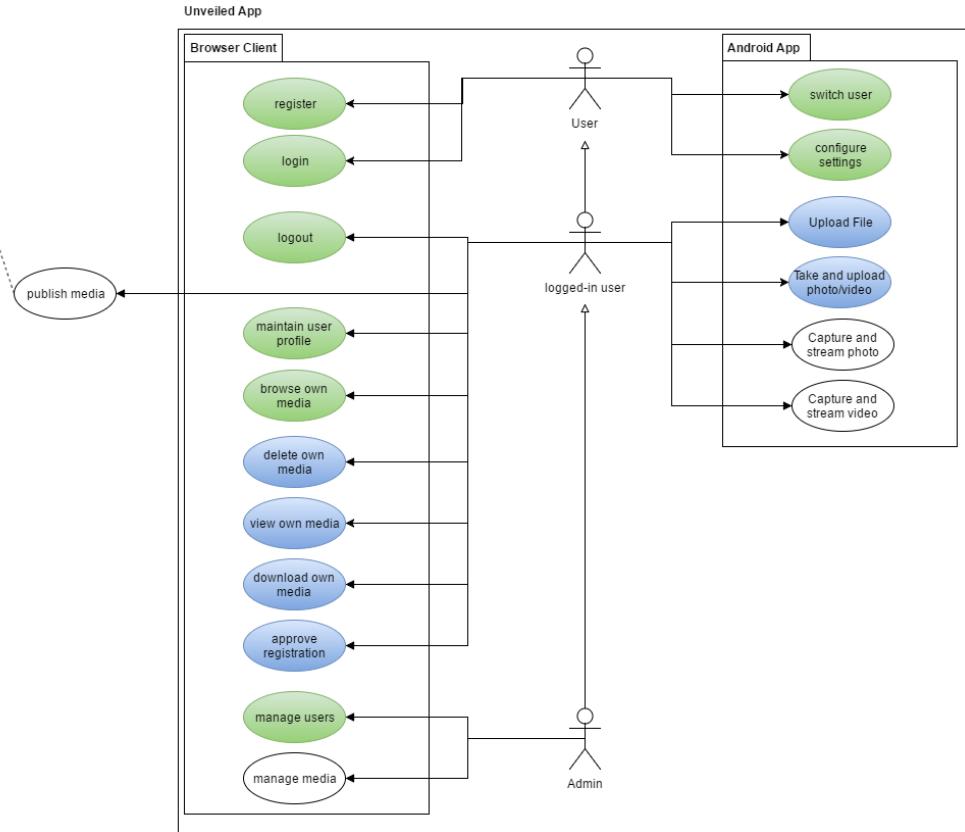
Software Requirements Specification (SRS)

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)

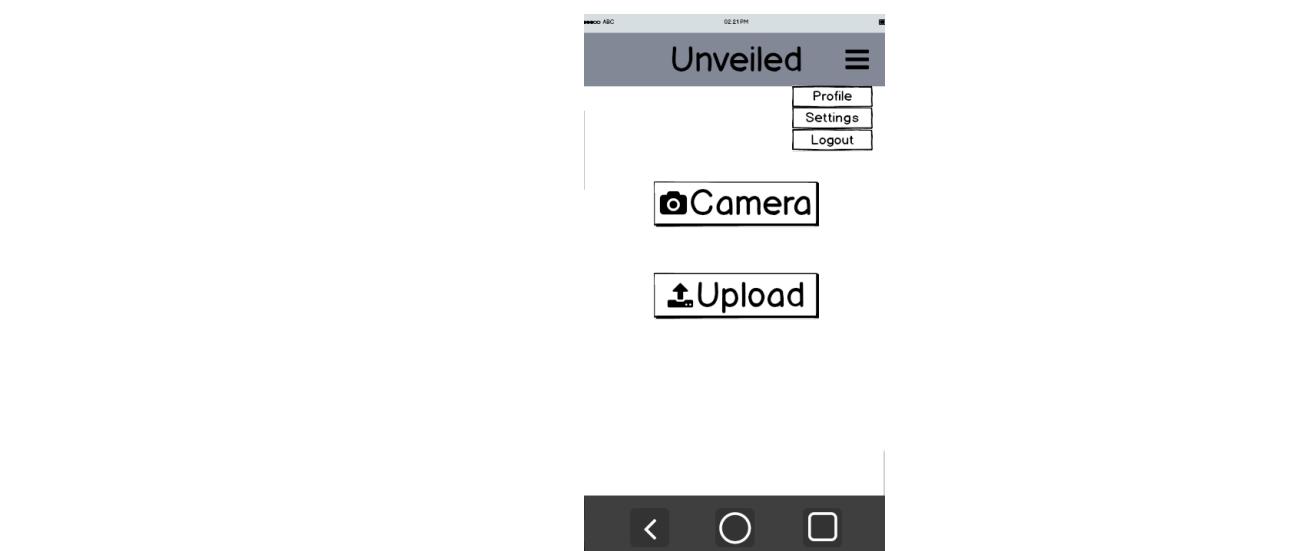
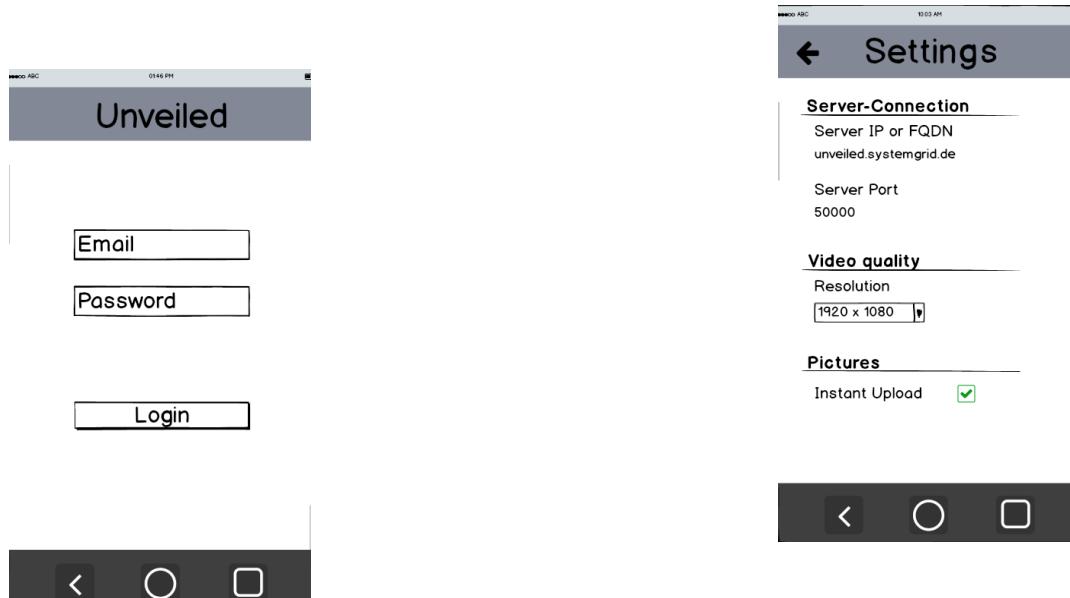
The following picture shows the latest version of our OUCD:



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/>.

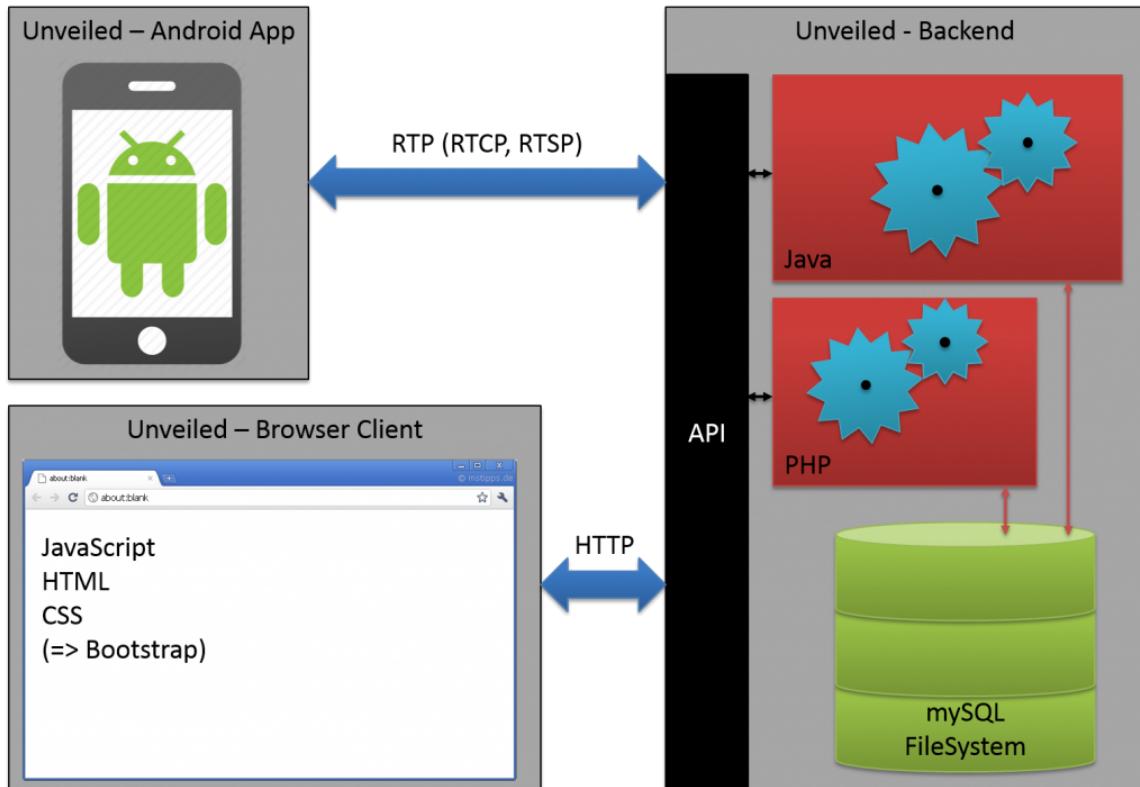


5. October 2015 by Digister on Allgemein • Edit→

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

O

[← 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