

The requirements have been divided into the FURPS+ model categories. For the requirements where it's applicable we use short user stories to describe the requirement. In other cases we use "The system shall...". The requirements in each category are prioritized based on the value for the product and how difficult they might be to implement.

In our iteration plans the individual requirements will be referred to with the ID they've got here in the product backlog.

Example: F1 refers to the first requirement in the Functionality category.

Google Docs

We have a Google Docs sheet for our product backlog so that all members of the group can do changes in the same document simultaneously. Having the product backlog as a Google Docs makes it easier for us to comment and prioritize the requirements. The product backlog is then transferred to our wiki product backlog once a week or at least in the end of every phase of the project (inception, elaboration, construction, transition). The transfer is done by downloading the Google document as a webpage (HTML), opening the HTML-files in an IDE and copy the tables. The copied tables are then converted from an HTML-table to a markdown-table using [markdownTables](http://markdowntables.mrvautin.com/) (<http://markdowntables.mrvautin.com/>).

Product backlog

(https://docs.google.com/spreadsheets/d/1fQBQyhCi1xqPi2szUyqQn_dRg4coN8qts=5ac73808#gid=0).

New in Google Docs (as of 16th of April)

In our Google Docs product backlog, in the Functionality category, we have re-prioritized the basic requirements for the application. We and the customer decided that we should focus on requirements that are critical for the regular user of the application. Therefore some requirements, that previously were listed with the other functional requirements, are listed under "In terms of time" and can be seen as bonus features.

The priority number goes from number one to five, with five being the highest priority. In addition to putting high priority on the requirements that are critical for the regular user, we have also focused on risk and dependencies. The requirements that are the most basic for the application to work, i.e. the ones with the highest risk, have the highest priority. If one requirement is dependent on another, the requirement with dependees needs to be implemented first, and therefore has a higher priority number.

Functionality

ID	Name	Description/User story	Dependencies	Status	Priority	Author
F1	User register	A user must be able to register to the app.	–	Implemented	5	Linda/Sofia
F2	User login	A user must be able to login to the app.	F1	Implemented	5	Sofia
F3	User – own data access	A user should have access to their own personal information.	F1, F2	Implemented	5	Andrew
F4	Add friends	A user should be able to add friends to a friendslist.	F1, F2	Implemented	5	Sofia
F5	User chat room	A user should be able to create chat rooms to which other users can connect.	F1, F2	Started	5	Linda
F6	User text message	A user can send text messages to other users.	F1, F2	Started	5	Andrew
F7	Logs through UI	The communication log is available to the user through the app's UI.	F1, F2	Not implemented	5	Andrew
F8	User camera photo message	A user can take a picture with the device camera and send it to other users.	F1, F2	Not implemented	4	Andrew

F9	User stored image message	A user can choose an image stored on the device and send it to other users.	F1, F2	Not implemented	4	Andrew
F10	User video call	A user can call another user and communicate with video.	F1, F2	Not implemented	4	Andrew
F11	User support	The user can contact admin via the UI for support.	F1, F2	Not implemented	4	Linda
F12	Admin/user chat	The admin can chat with users.	F1, F2	Not implemented	4	Linda
F13	Consent Mail	Before a user begins using the app, the user receives a mail with a link. Clicking upon the link allows the user to consent to use the app.	F1	Not implemented	4	Andrew
F14	User consent withdrawal	The user can at any time withdraw their consent via the app's UI.	F1, F2	Not implemented	4	Andrew
F15	Consent withdrawal message	If consent is withdrawn, the user receives a message confirming that the user will be removed from the system and that all records of communication will be removed.	F1, F14	Not implemented	4	Andrew
F16	Consent withdrawal confirmation	When the user receives such a message, the user can confirm removal from	F1, F14, F15	Not implemented	4	Andrew

		system.				
F17	User removal upon consent withdrawal	If consent is withdrawn, the user is removed from the system within 24 hours.	F1, F14, F15, F16	Not implemented	4	Andrew
F18	User data removal upon consent withdrawal	If consent is withdrawn, all records of communication are removed from the system. Including the chat logs of other users where the user has participated.	F1, F14, F15, F16, F17	Not implemented	4	Andrew
F19	User account inactivation	It should be possible for a user's account to be inactivated (unclear if the user can do this or admin or both?)	F1	Not implemented	3	Andrew
F20	User account deletion	It should be possible for a user's account to be terminated (unclear if user does this or admin or both)	F1	Not implemented	3	Andrew
F21	Log erasure notification	The user is advised upon signup that logs will be erased after a period of time.	F1	Not implemented	3	Linda
F22	Log storage time	Logs should not be stored for a longer period of time than required by law.	F1	Not implemented	3	Andrew
		Each user should				

F23	Registered relative	have a registered relative	F1	Not implemented	2	Linda
F24	Registered relative can withdraw consent	The registered relative should be able to withdraw consent on behalf of the user	F1	Not implemented	2	Linda
F25	Registered relative can erase users account	The registered relative should be able to erase the users account	F1	Not implemented	2	Linda
F26	User channel creation	A user can create a channel and send live from her/his camera.	F1, F2	Not implemented	1	Andrew
F27	User channel invite	A user can invite other users to watch the channel by copying a link and distribute that link to the viewers.	F1, F2, F26	Not implemented	1	Andrew

Usability

ID	Name	Description/User story	Dependencies	Status	Priority	Author	Topic
U1	Simple UI	The application should have a clear and simple user interface.	–	Started	5	Andrew	–
U2	WCAG 2.1	The design should be in accordance with WCAG 2.1 – minimum level AA, better if possible.	–	Not implemented	5	Andrew	–
		The user interface should make use of well-defined					

U3	UI buttons	buttons based on available actions – for instance writing a new message, taking or uploading a photo, starting a video conversation.	–	Started	4.75	Andrew	–
U4	Responsive	The UI is reponsive and works on mobile and tablet devices.	–	Started	4.5	Andrew	–
U5	App accessibility	The app should be accessible for all users with access to the internet regardless of the device being Android or iOS.	U1	Not implemented	4.5	Andrew	–
U6	UI extents	The full extent of the user interface should be utilized by the user’s ongoing communication.	–	Not implemented	4	Andrew	–

Reliability

ID	Name	Description/User story	Dependencies	Status	Priority	Auth
R1	Environment	The app should work on smartphones and tablets.	–	Started	5	Andre
R2	Harmful code	The system should be protected from harmful code, specifically	S1	Not implemented	4.75	Sofia

		OWASP Top 10.				
R3	Consistency	The app should have consistent functionality across all operating systems and environments.	–	Started	4.5	Andre
R4	Encrypted text on server	All text information stored on server, such as communication logs and personal data, should be encrypted.	–	Not implemented	4.5	Linda
R5	Secure text on server	All text information stored on server, such as communication logs and personal data, should be protected from access by third parties.	–	Not implemented	4.5	Linda
R6	Secure communication	Communication should take place securely, so that no third party can access transmitted data.	–	Not implemented	4.25	Andre
R7	Secure image storage	Images should be stored securely so that no third party has access to them.	–	Not implemented	4	Andre
R8	Virus protection	The system should be protected from viruses – OWASP Top 10 mentioned by customer.	S1	Not implemented	4	Andre

R9	Limited client data storage	As little data as possible should be stored on clients in order to prevent unwarranted spreading of pictures, films and other information.	–	Not implemented	3.5	Andre
R10	System backup	All information on system should be backed up in a second location so that it can be restored in case the data in the first location is compromised	–	Not implemented	3	Andre

Performance

ID	Name	Description/User story	Dependencies	Status	Priority	Author	Test case
P1	Stress tolerance	The system should accommodate at least 100 simultaneous users (both solely logged in and active) without noticeable detriment to the systems functionality, accessibility or usability.	–	Not implemented	4	Andrew	–

Supportability

		Description/User					Test
--	--	------------------	--	--	--	--	------

ID	Name	story	Dependencies	Status	Priority	Author	cas
S1	System protection updates	The system's protection against viruses and harmful code can be updated	R2,R8	Not implemented	3	Andrew	-