Design and Use Cases

Introduction

In this document we describe the core functionalities of the health system we are planning to develop. It contains all the information collected so far to start detailed planning of the system. Besides the definition of the core functionality of the system in the first chapter of this document we divided up the process of the realization of the system in different phases and discussed possible steps to be done in each phase. Then we illustrate first ideas for the design to use for every phase and compare certain aspects of the system to the realisation done in the MIT project Indivo. In an additional chapter we discuss further ideas for the system which lay outside the core functionalities and shall use the API and extension facility to integrate into the system. Finally we list all the use cases in decreasing order of importance for the system according to the splitting in the different phases.

Contents

Core Functionalities

The following list contains entries which are according to our point of view essential to an electronic health system. All additional functionalities may then build upon these core functionalities. The word institute is used in the following to refer to a group of institutes like doctors, doctor's offices, hospitals, dentists, physiotherapists, psychologists, special clinics and so on.

- Users can create a user account and using this data they can log in and log off the system. This feature allows users to quickly find and identify their data in the system and protect it from others eyes.
- Users can store, edit and delete their personal health records in their systems account. This allows full control of a person's own copy of their health record anytime and anyplace.
- Institutes can create an account and log in and off the system. In their account they can provide detailed description about their area of expertise, business hours and whatever they feel useful for the users to know. This helps institutes to be seen by user and for advertisement.
- A permission system allows users to make their health record or parts of it visible to other users and/or to institutes. This allows family, friends and institutes to see and possibly edit records of users.
- Provided the user granted access, institutes can inspect and create new records in the health record of a user. This allows e.g. a doctor to upload all the relevant information about the latest visit to the user account or gives a hospital insight into important information after an accident happened.
- A record in the user health records shall be able to contain files (or pointers to) such as X-ray photographs, MRI, etc. next to all kind of attributes and text.
- A search engine allows to search for users, institutes, records (within a user account), extensions and other parts of the system. This functionality helps to navigate in the system and to find the right institute in case of specific health problems.
- An API provides access to all the core functionalities as well as to possible extensions to the core. Besides it allows extensions to the core to store extension specific data. This functionality allows the system to grow and external partners to integrate and be part of the system.

Of course there are many more features that such a system could and shall provide. As they all build to a certain extent on top of these core functionalities they are not part of this list but will be discussed in the remainder of this document.

Implementation Order of Core Functionalities and Certain Extensions

In this section of the document we like to discuss the functionalities in more detail and decide which of them will have what priority when it comes to the implementation. To do so we split the process of realisation of the system in several phases. To each phase we assign certain functionalities which shall be completed when the phase ends. A phase should represent the amount of work realisable in roughly one to two weeks. It goes without saying that the phases are ordered in descending order of importance for the system.

Phase 1

In this stage the core of the core functionalities as listed below shall be realized in order to have something to work with.

- Users shall be able to register themselves and log in and out of the system
- Institutes shall be able to register themselves and log in and out of the system
- Users shall be able to create basic health records and add them to their health record
- Users can inspect their records in a list view and detailed view
- All data gets stored persistently and shall be fast accessible.

Phase 2

Once the system is up and running we can extend the existing functionalities and extend them with new features.

- Users shall be able to edit and delete information on and their user account itself.
- Institutes shall be able to edit and delete information on and the account itself.
- Health records shall be able to contain more information as well as files from X-ray etc.
- The system shall provide a basic search engine to find other users and institutes. For other users only the name and maybe an image, but no further information shall be shown.
- Users shall be able to allow institutes to access their health record (All or nothing for now).
- Users shall be able to see in a separate view a list of all institutes with access rights to their health records.
- Institutes shall be able to create new records and add them to a user's health record.
- Records shall be editable and the created shall be able to delete them.

Phase 3

In this phase we lay more focus on the permission system and on grouping different things together. Thereby the principle of access categories shall be realised in a similar manner then the Circles Google introduced in Google Plus.

- Users shall be able to assign each and every entry in their record to one or more access categories.
- In a separate view (the one discussed in phase 2) the user shall be able to create access categories and assign other users and / or institutes to these categories. Users shall further be able to edit and delete categories and change the assignments of users and institutes.
- Institutes and users shall be able to inspect all the record entries of another user if this particular user assigned them and the respective entries to the same access category.
- A user shall be able to group certain entries of his or her record into folders (e.g. all entries for a certain injury). The access category given to the folder then applies to all containing records. Users shall be able to put the same record in multiple folders.
- Institutes shall be able to create access categories as well to hide certain information to e.g. everyone but customers and to keep something like a friend list.

Phase 4

Having realised all the core functionalities it is now time to create an API to access these functionalities and to expand the functionalities.

- Part of the API shall allow to log a user into the system from an external application and to give an external application the rights to access certain data.
- There shall be a new category of accounts in the system similar to institutes for external applications which describe the application. The more it shall illustrate the user what data this application needs access to.
- Users shall be able to see and change the access categories for external applications in the respective access view. The more they shall be able to “uninstall” an application and delete all the generated data by this application.
- Part of the API shall allow an external application to add new specific data types to the system and once this is initialized also to a user’s account (data which does not have to be a record or personal data such as information about the last jogging run).
- Via the account for applications it shall be able to register the application to the system and make it accessible for users. The data for the application shall be editable and it shall be possible to delete application while the user can decide if he or she wants to keep the so generated data. An interface shall allow updates for the applications without data loss.

Phase 5

Having now the possibility to extend the system we want to integrate new functionality.

- Users shall be able to integrate records of their ancestors to indicate possible genetic diseases.
- A messaging system shall be added to the system which allows users, institutes and applications to send email-like messages to each and another.
- The search engine shall be expanded to find applications and messages besides users and institutes.
- The API shall be expanded to allow access to the messaging system and the search engine.

Phase 6

As a next step we want to integrate research possibilities into the system.

- There shall be a new account type for researches similar to institutes accounts. Public accessible data in these accounts shall include area of research, a description about current topics and links to more information.
- Research accounts shall be able to create surveys and announce experimental studies on their accounts.
- The search engine shall allow finding research accounts.
- The messaging system shall be expanded for research accounts.
- Users shall be able to add research accounts to their access categories and thereby allow them to inspect (not edit) their data for research. This shall be an all or nothing decision meaning that all the research accounts shall be able to view whatever category the user assigned them to. The more a user shall be able to give single research accounts more access (but not less).

- The search engine shall be expanded with a query engine that allows a research account to ask queries beyond a single user. This shall allow finding different people with a certain disease or injury.

Phase 7

Depending on the time left this phase shall be devoted to a health platform providing the users all health related information needed.

- The system shall be expanded with a health platform that provides the latest health news.
- The health platform shall contain e drugs register including side effects and detailed description.
- The health platform shall contain a wiki for all kind of injuries and diseases.
- The health platform shall contain nutritional advices.
- Institutes and research accounts shall be able to edit the health platform entries including discussions on certain topics and versioning (similar to Wikipedia).
- Advertisements for institutes, research topics and extensions shall help to make them more aware for the user. Advertisements shall be only on the health system and kept in a way in which they do not disturb the user experience of the system.

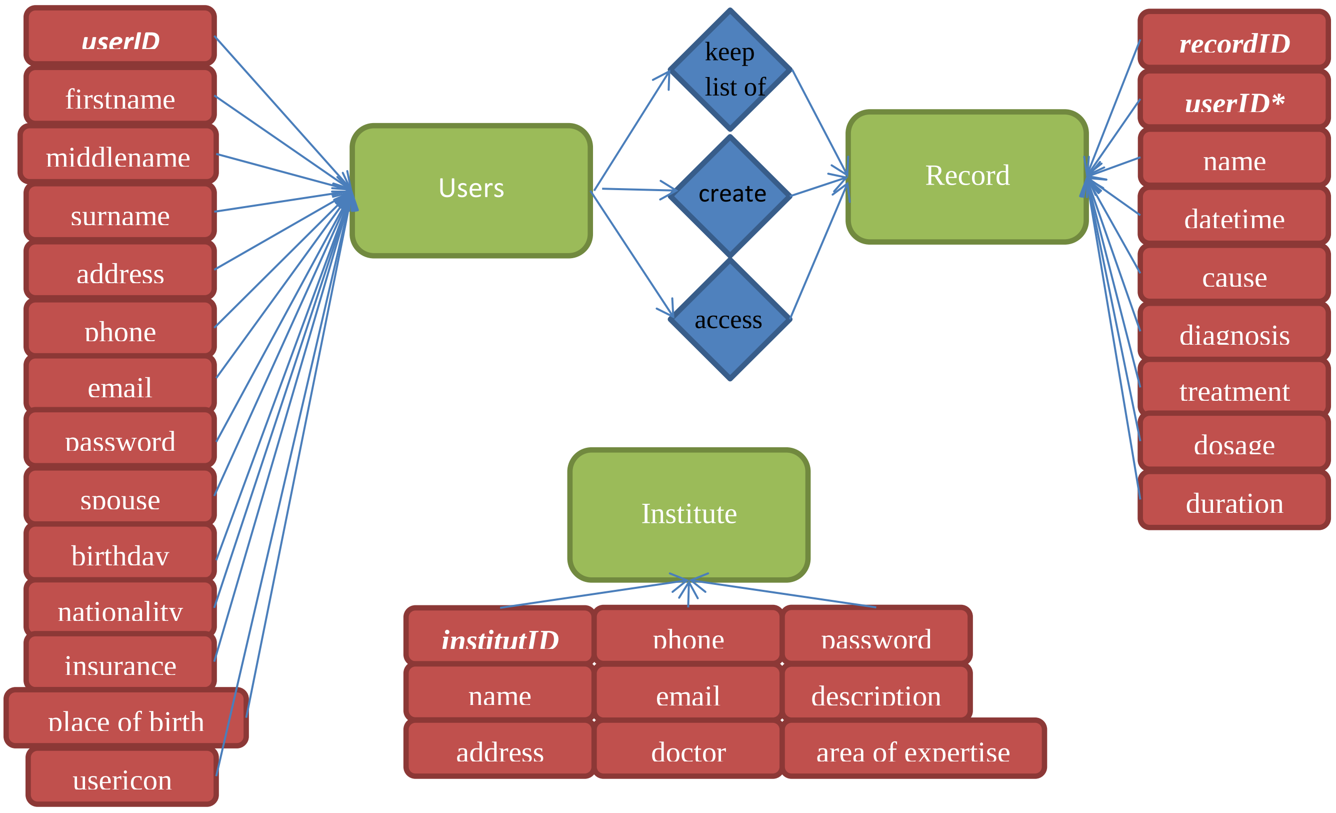
Phase 8

Starting at this point it shall be time for all kinds of external application to include in the system. A list of possibilities is given in the chapter “Ideas for Additional Extensions to the System” later in this document. Besides from this list there are endless possibilities for the brilliant ideas of developers.

Design of the Different Implementation Phases

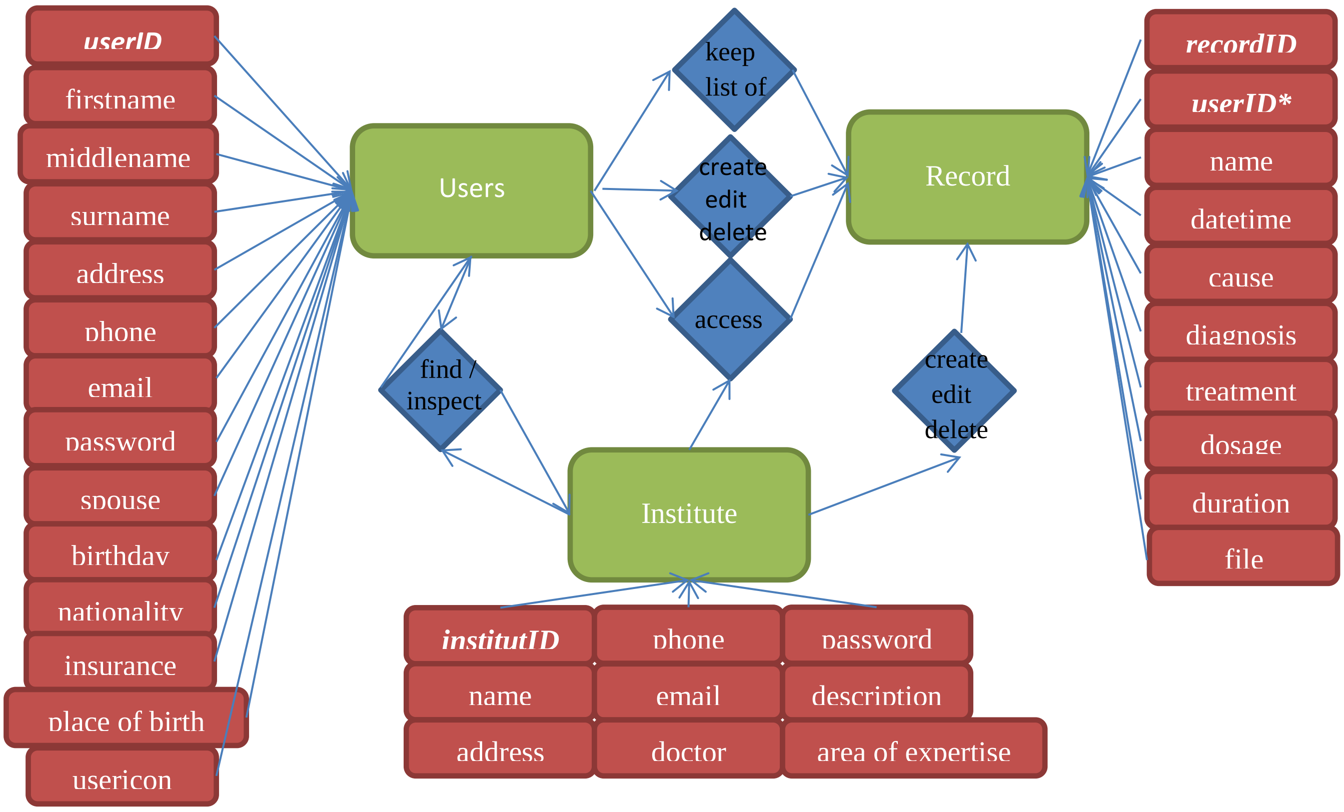
Keys are written bold, italic and underlined. Foreign keys additionally have a “*” at the end.

Phase 1



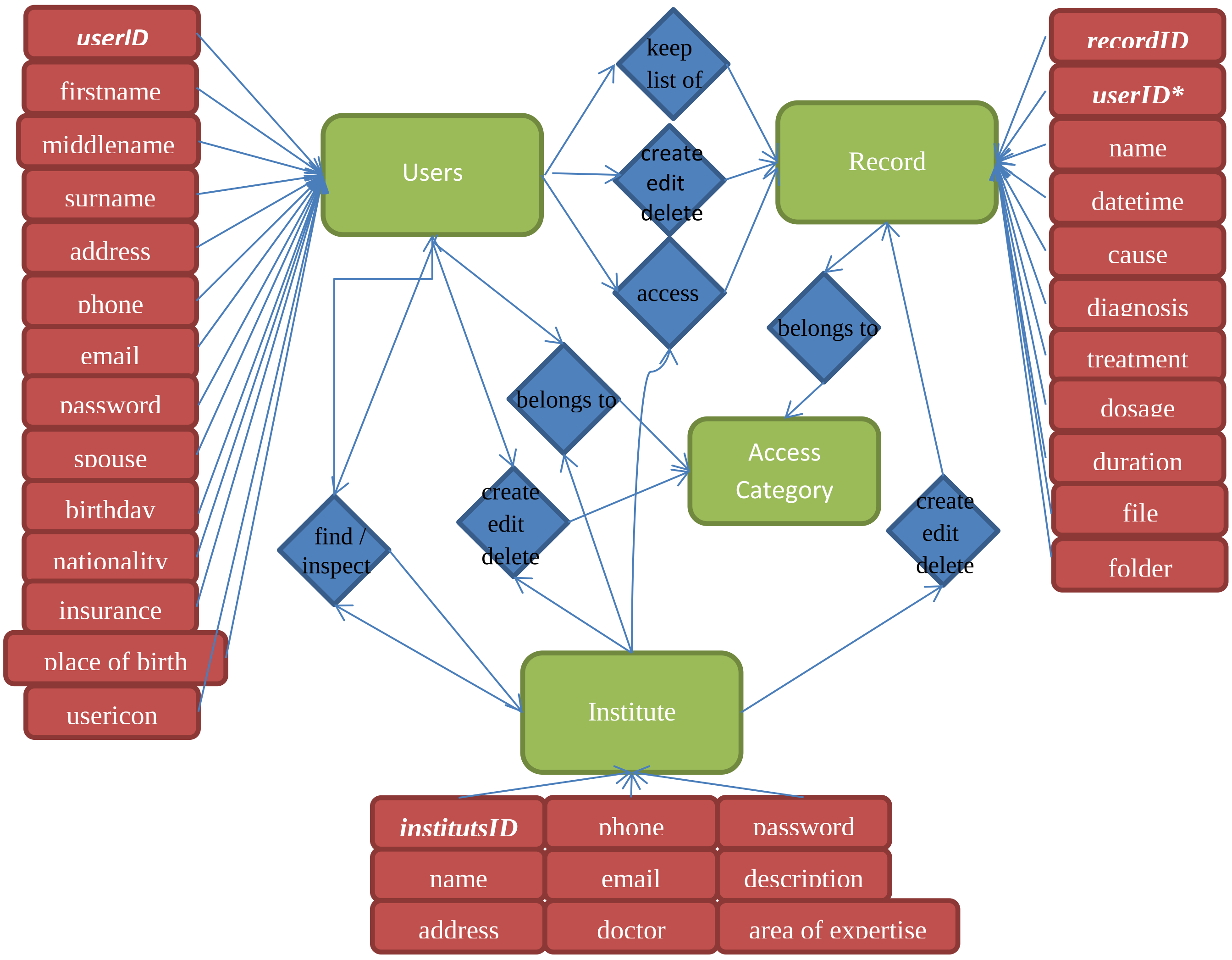
As defined in the previous chapter we do not need a connection between institutes, users and records just yet. Hence we can focus on building the data structure for the individual accounts, the login process for both kind of accounts and the session management. Also the records are not very specific yet, which allows us to lay the focus on how to store and assign them to different users. The idea behind this design choice for the first phase was that we can have an easy start with the new tools and especially the database and database integration. With the fact that we need access to the records and that records are assigned to users we can test the query functionality of the database as well. Once we master these tasks, we should make good progress with the next phases.

Phase 2



In this phase we now implement the link between institutes, users and their records. Access rights are also introduced in this phase. As they are rather hard to integrate in a diagram of this kind, we keep this for the diagram of the next phase. Probably record will get many more attributes over time. Since we do not know yet, what is effectively needed, we keep it the way it is for now.

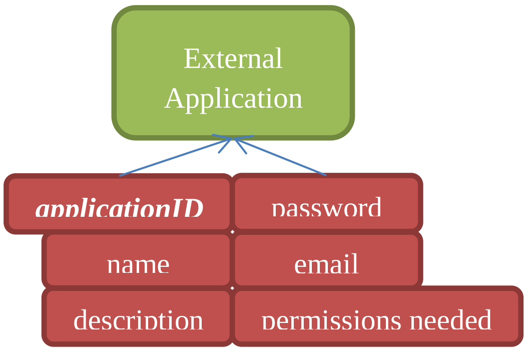
Phase 3



In phase 3 we introduce a more sophisticated permission system by introducing access categories. The more we introduce folders for records.

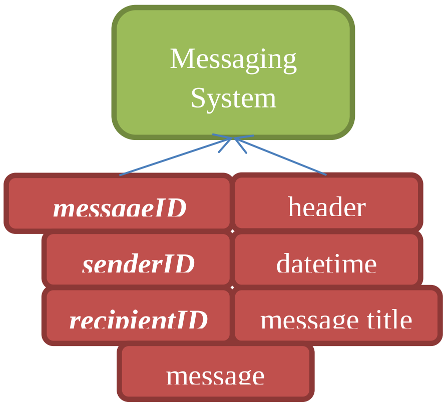
Phase 4

In this phase we introduce an API which accesses the functionalities defined above. For the sake of cleanness we do not even try to draw this in the drawing. But there is a new group of accounts in the system which might look like this:



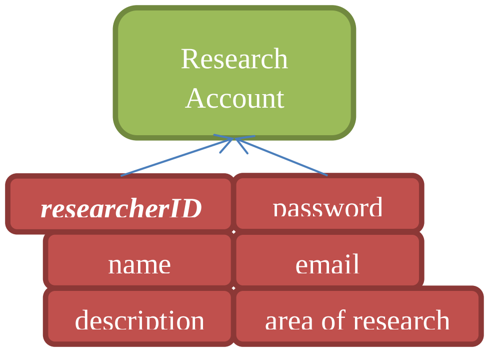
In this drawing we did omit the relation to access categories, access to records if allowed by the users, searching of applications and the like. But we are totally aware of the existence of these facts and consider it on detailed design.

Phase 5



In this phase besides others we introduce a messaging system. Both senderID and recipientID as noted in the diagram above are actually either userID's, instituteID's or applicationID's. Again we omit the relations to the participants in order to not blowing up this document too much.

Phase 6



Now we get to the research accounts. This is particularly the same as in phase 4 for the external apps. We omit again the relations to other users to save some space.

Phase 7

Since we do not know a lot about the health platform at this point and we are not even sure if we will find the time to implement it, we omit the drawing of a diagram for now. This will be done if the time is right.

Discussion on Indivo and its Functionalities

Most of the information mentioned below is taken from the documentation of Indivo v.2.0. Here is a link to the webpage of the documentation: http://docs.indivohealth.org/en/v2.0.0/index.html

- Indivo is written in Python.
- Authorization is done with OAuth.
- They provide a REST API for extensions.
- All communication is done via HTTPS.
- Background server – Indivo X server –to deal with accounts and medical records inclusive documents. This server is responsible for authentication and authorization and provides the UI for Admin and User applications.
- Indivo User Interface (Indivo Chrome) is the web based GUI for the system
- User applications can be added by users and what they are allowed to access from the system is user defined as well. They provide a web interface (usually via an IFrame).
- Admin applications can connect to the X server and create new accounts or change them. The more they can change ownership of records and record metadata in general. They do not have access to the content of records, only on the metadata.
- A record in Indivo consists of one or multiple documents and can be accessed by one or multiple accounts. Nevertheless there is one account defined as the owner of the record.

- There are different data models based on Django's ORM. A Query API and a Reporting API make these data models queryable.
- They provide SDMX Schema to define the format of incoming data in their SDMX specification language. The more schemas are used for XML validation. XSD is used.
- The pipeline for incoming data is illustrated in figure 1. As mentioned before this data has to be in XML format and has to agree to a XSD schema. Then it is transformed according to a transform output format and converted into one or more Fact objects for storage. Via reporting API one can then query the database for matching facts. These facts will be serialized to XML or JSON for the app to retrieve.
- A fact is a single data point e.g. one medication formatted according to a data model. In our e.g. this would be the medication data model.
- Indivo provides a messaging and notification system similar to emails. Messages have a subject and a body. Messages can be sent to records as well where they will be rerouted to the owner and all accounts that have full access to that record.
- To register an app with Indivo an app manifest is needed. The SMART Project's syntax is used for this with some Indivo-specific parameters. In the manifest the app can define among others the Indivo version to use, whether it can be used in an iframe or not, if it has an UI and the oAuth callback URL. The more there are the usual facts such as name, description, version, icon, etc.

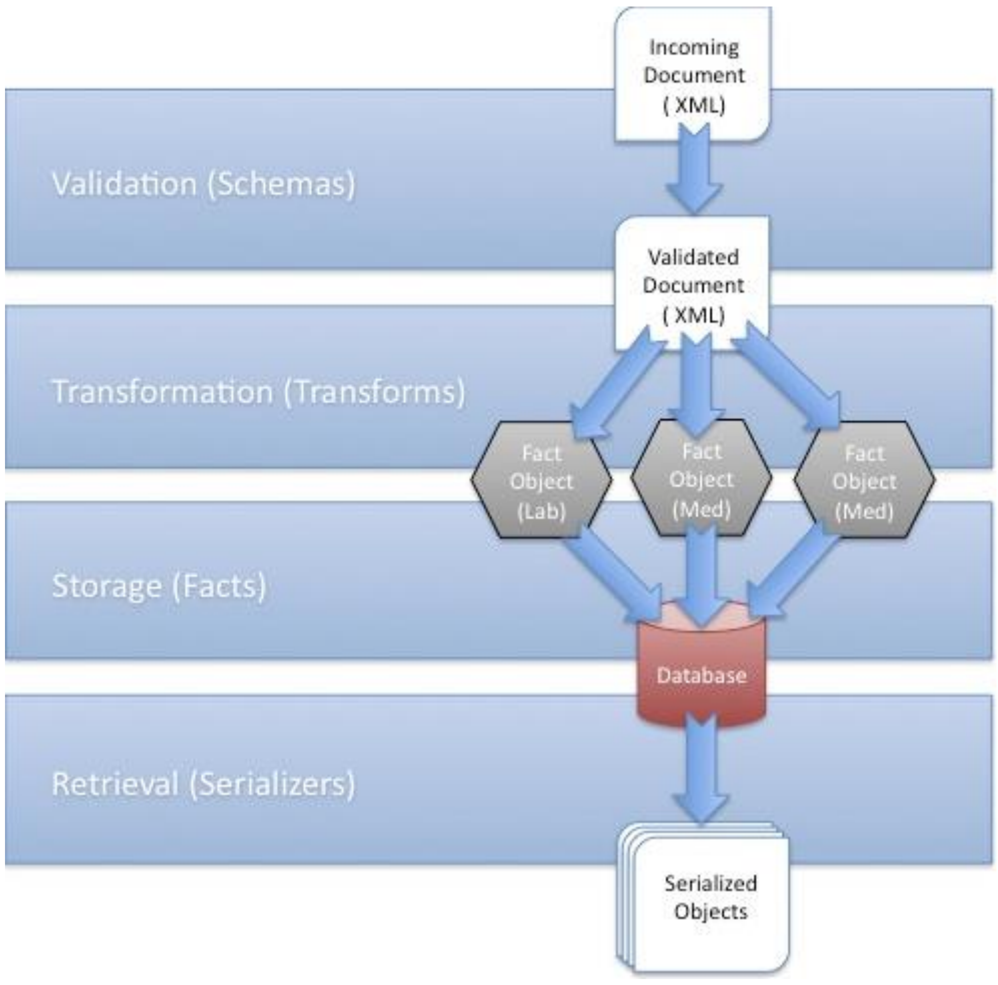


Figure 1: The Indivo Pipeline

Ideas for Additional Extensions to the System

- User interfaces for hospitals and doctor's offices that help their administrative stuff to easily connect to our system and upload the patient's data.
- Integration of the eHealth and ePatientendossier applications in Switzerland and other countries. These applications contain information saved on the insurance cards of individuals. A good integration to our system would be e.g. to provide access to the users account on our system using his or her insurance card in case of an emergency.
- The process of organ donation is rather complicated and dealt with in a separate database. Integration of the Swisstransplant (and similar in other countries) would possibly help to find new donors and ease up the process, since all necessary information is stored in a single database in our system.
- Integration of the health insurances would provide them with the data they need and allow the user to find the perfect fit for him. An application could be done for example in association with Comparis or similar health insurances comparison companies. Using the query engine such an application could analyse the records of a user and provide him or her with tips regarding finding the perfect insurance company.
- An integration of e.g. the Rega, the paraplegic centre in Nottwil or any other health trust would help them to find more people and again combine different databases.
- The possibility for institutes to synchronize or at least publish their calendar on the system would probably help users to find an appointment and the institutes to have a tighter schedule.
- A rating system for institutes would help users to find the doctor they look for. In addition bad experiences or of course compliments could be written in comments next to the ratings. People like to give feedback and hopefully this will increase the services in many institutes since else they might get a lower rating.
- The search engine could be extended by a location based services to find local institutes that match the need of a user. The advertising system could profit from such a service as well and it would increase the experience for the user.
- Integration of chemist's shops together with saving prescriptions by doctors in the system would allow users to have everything online and accessible all the time. This would erase the problem of losing prescriptions on the way to the chemists.

Use Cases

This document shall give a first impression of possible use cases for the health platform. First we introduce some general use cases, which might include several actors and interfaces, called "big picture use cases". Then we go more into detail and explain use cases for different actors and different parts of the system.

All the use cases in the remainder of this document are composed of the following components:

- **Use Case Name:** A short name for the use case. Usually given as the title of the paragraph
- **Use Case ID:** An unique identification number within this document
- **Application:** Which part of the application or system does this pertain to
- **Primary Actor:** Who is the main actor in this use case
- **Description:** A short description of the use case

- **Priority:** How important is the realisation of this use case (Options include: High, Middle, Low, Optional)
- **Precondition:** What has to be met before this use case can start
- **Trigger:** What event will trigger this use case
- **Flow:** The events of the use case chronologically listed for the case when everything goes without errors or exceptions
- **Alternative Flow:** The most important alternatives to the flow described above including errors and exceptions

Use Cases for Users

Register a User

ID: 1.1
Application: Login/Register
Primary Actor: User A
Description: Create and register a new user to the system
Priority: High (Phase 1)
Precondition: The system is up and running
Trigger: Person A clicks the “Register”-button
Flow:

1. The registration form is shown on the screen witch asks for full name, email and password
2. Person A fills out all the forms and hits the “Register”-button
3. The system creates a new person entry and redirects the user to the login screen

Alternative Flow:

- In 3: If Person A’s input is erroneous or there was an error while creating a new user an error message gets displayed

Login as User

ID: 1.2
Application: Login/Register
Primary Actor: User A
Description: Add username and password to log into a user account of the system
Priority: High (Phase 1)
Precondition: The system is up and running. A has registered an account on the system
Trigger: A clicks the “Login” button on the starting page
Flow:

1. In a dialog A enters username/email and password
2. A hits the “Login”-button
3. The system checks if the provided data is correct and starts a new session
4. A is led to news page of health platform

Alternative Flow:

- In 3: If the provided data is incorrect an error message is displayed
- In 4: If there is no health platform yet, the user gets redirected to the personal record page

Logout as User

ID: 1.3
Application: Logout

Primary Actor: User A
Description: Log out of the user account of the system to end the session
Priority: High (Phase 1)
Precondition: A is logged in
Trigger: A hits the "Logout"-button anywhere on the system
Flow:

1. A dialog opens to ask A if he/she really wants to logout
2. A hits the "Yes"-button
3. The system stops the session and redirects A to the start page

Alternative Flow:

- In 2: A hits the "No"-button and the dialog is dismissed

Add Consultation to User Record

ID: 1.4
Application: Add Record
Primary Actor: User A
Description: Have you had to visit a doctor? Add the consultation record to your personal record.
Priority: High (Phase 1)
Precondition: A has a user account and is logged in
Trigger: A hits the "Add Record" – button on the User Record page
Flow:

1. A hits the "Add Consultation" button from the list of different options
2. A form with all possible information about a consultation is displayed
3. A fills out all the form fields
4. A clicks the "Add"-button
5. The system saves the new entry to A's personal record
6. A is redirected to the User Record page where the new entry appears on the top

Alternative Flow:

- In 1: If A hits another button, see the following use cases
- In 5: If A's input was illegal or some important information was missing as well as if there happened an error while saving an error message is displayed

Inspect Personal User Record

ID: 1.5
Application: User Record
Primary Actor: User A
Description: A can anytime visit his personal record page, where all consultations/hospital visits/surgeries/medicine/prescriptions/etc. are visible
Priority: High (Phase 1)
Precondition: A has a user account and is logged in
Trigger: A hits the "Record"- Link
Flow:

1. A list with all the records saved for A is displayed in chronologic order with the latest entry on top

Alternative Flow:

- In 1: If there are no entries yet a message to inform A is shown instead

Change Personal Settings

ID: 1.6
Application: Personal Settings
Primary Actor: User A
Description: Change any kind of personal settings of your account like address, phone number, email, etc.
Priority: High (Phase 2)
Precondition: A has a user account and is logged into the system
Trigger: A clicks on the “Settings”- Link
Flow:

1. The settings page with a form for the different user settings such as name, address, phone number, email and system specific settings is shown
2. A adds or changes value(s) in the form
3. A hits the “Save”-button
4. The system stores all the values to the user account
5. A success message is displayed

Alternative Flow:

- In 1 or 2: A clicks any other link or button on the page. A dialog asks A if he wants to discard the changes. If A clicks yes, he/she is led to the clicked page, else the dialog is dismissed again
- In 4: If there happened to be an error while saving the data or A provided illegal input an error dialog is displayed

Add Hospital Stay to User Record

ID: 1.7
Application: Add Record
Primary Actor: User A
Description: Did you have to stay at a hospital? Add all the information about it to your personal record
Priority: High (Phase 2)
Precondition: A has a user account and is logged in
Trigger: A hits the “Add Record” – button on the User Record page
Flow:

1. A hits the “Add Hospital Stay” button from the list of different options
2. A form with all possible information about a consultation is displayed
3. A fills out all the form fields
4. A clicks the “Add”-button
5. The system saves the new entry to A’s personal record
6. A is redirected to the User Record page where the new entry appears on the top

Alternative Flow:

- In 1: If A hits another button, see the use cases 1.4 and 1.8, 1.9
- In 5: If A’s input was illegal or some important information was missing as well as if there happened an error while saving an error message is displayed

Add Surgery to User Record

ID: 1.8
Application: Add Record
Primary Actor: User A
Description: Add all information as well as files like MRI/X-ray to your personal record
Priority: High (Phase 2)
Precondition: A has a user account and is logged in
Trigger: A hits the "Add Record" – button on the User Record page
Flow:

1. A hits the "Add Surgery" button from the list of different options
2. A form with all possible information about a consultation is displayed
3. A fills out all the form fields
4. A clicks the "Add"-button
5. The system saves the new entry to A's personal record
6. A is redirected to the User Record page where the new entry appears on the top

Alternative Flow:

- In 1: If A hits another button, see the use cases 1.4 and 1.8, 1.9
- In 5: If A's input was illegal or some important information was missing as well as if there happened an error while saving an error message is displayed

Add Medicine Entry to User Record

ID: 1.9
Application: Add Record
Primary Actor: User A
Description: Add drugs you take and prescriptions to your personal account. See use case 1.10 for prescriptions
Priority: Medium (Phase 2)
Precondition: A has a user account and is logged in
Trigger: A hits the "Add Record" – button on the User Record page
Flow:

1. A hits the "Add Medicine" button from the list of different options
2. A form with all possible information about a consultation is displayed
3. A fills out all the form fields
4. A clicks the "Add"-button
5. The system saves the new entry to A's personal record
6. A is redirected to the User Record page where the new entry appears on the top

Alternative Flow:

- In 1: If A hits another button, see the use cases 1.4 and 1.8, 1.9
- In 5: If A's input was illegal or some important information was missing as well as if there happened an error while saving an error message is displayed

Change Entry in User Record

ID: 1.10
Application: Add Record
Primary Actor: User A

Description: Any entry which was added by A can be changed later on to correct possible errors in the data
Priority: Medium (Phase 2)
Precondition: A has a user account, is logged into the system and has at least one entry in the user record
Trigger: A clicks on the “Edit”-button next to an entry in the User Record page
Flow:

1. Depending to the type of the entry (see use cases 2.5-2.8) the according form is displayed and filled with the saved values
2. Continue with 3. in the according use case from 2.5-2.8

Alternative Flow:

Delete Entry in User Record

ID: 1.11
Application: User Record
Primary Actor: User A
Description: If A entered a record completely wrong, it is possible to delete it from the system
Priority: Low (Phase 2)
Precondition: A has a user account, is logged into the system and has at least one entry in the user record
Trigger: A clicks the “Remove”-button next to an entry in the User Record page
Flow:

1. A dialog is displayed asking A if he/she is sure and really wants to delete the entry
2. A clicks the “Delete”-button
3. The system marks the entry as deleted
4. The dialog content is changed to a success message and disappears after 3 seconds. The entry is removed from the User Record page
5. On the next clean-up of the system database the entry gets deleted

Alternative Flow:

- In 2: A clicks the “No”-button and the dialog is dismissed
- In 3: The system might delete the file right away

Search

ID: 1.12
Application: Search
Primary Actor: User A
Description: In the search page A can look up other people, find information about doctor offices and hospitals as well as about research people. The more, external apps can be found via this page. The search page shall allow finding almost everything on the system similar to a Google site related search
Priority: Medium (Phase 2)
Precondition: The system is up and running, A has a user account and there is some data in the system
Trigger: A clicks on the search box of a page or navigates to the Search page
Flow:

1. A enters the name of another user/doctor's office/hospital or application or parts of the name
2. A clicks enter on the keyboard or the search button
3. The system searches its records to find matches
4. A list with matches is displayed to A with "most important" on top

Alternative Flow:

- In 4: If there is no information found to the provided search term, a message is displayed stating this fact and proposing other search terms

View Records of other Users

ID: 2.13
Application: User Record
Primary Actor: User A
Description: Person A can inspect a Person B's page. Depending on what kind of data B allowed A to see, this page will show more or less content
Priority: Medium (Phase2)
Precondition: A has a user account and is logged in. There are other users registered to the system
Trigger: A clicks on the user image or name of another user
Flow:

1. The system loads the data records of user B which are associated to the same circle as the circle defined by B where B added A into and shows them on the screen

Alternative Flow:

- In 1: If A is not in a circle of B, the system displays all the information about B which B declared as publically visible

Create Access Categories

ID: 1.14
Application: Access Category Manager
Primary Actor: User A
Description: In the Circle Manager it shall be possible with one click to add a new circle and give it a name. Every person B added to a circle may see whatever content A approved for this circle
Priority: Medium (Phase 3)
Precondition: The system is up and running, A has a user account and is logged in
Trigger: A clicks the "Add Category"-button on the Access Category Manager
Flow:

1. A dialog opens that asks for a name
2. A enters the name for the new category in the text field
3. A clicks the "Add"-button
4. The system creates a new category and updates the view to display the new category along with the possible existing ones. The dialog is dismissed

Alternative Flow:

- In 2,3: A clicks the "Cancel"-button to dismiss the dialog and cancel the creation of a new category
- In 4: If the input was not correct or the system encountered an error, a message will be displayed

Add Person to a Category

ID: 1.15

Application: Access Category Manager

Primary Actor: User A

Description: A shall find other people or doctors in the Access Category Manager and add them to one of the before created categories

Priority: Medium (Phase 3)

Precondition: A category exists and A has a user account on the system. There is a second person B in the system

Trigger: A clicks on the category dropdown menu next to another user or a doctor

Flow:

1. On the dropdown list all access categories are listed next to a checkbox
2. A checks the desired checkboxes next to the categories
3. A clicks the “apply”-button at the end of the dropdown list
4. The system saves the changes

Alternative Flow:

- In 1,2: A clicks somewhere on the page next to the dropdown menu and the menu disappears
- In 2: A clicks the “Cancel”-button and the dropdown menu disappears
- In 4: If there happened an error while saving, a message gets displayed

Change Record Assignment to Categories

ID: 1.16

Application: User Record

Primary Actor: User A

Description: Each record shall be allocated to at least one circle. A shall be able to change this allocation user record listening

Priority: Medium (Phase 3)

Precondition: A has a user account and entered a data record to his/her personal record. A has defined at least one category

Trigger: A clicks on the dropdown menu for the access categories next to a record

Flow:

1. The dropdown menu opens and shows a list of all user defined categories.
2. A checks the boxes next the categories he or she wants to select for this record
3. A hits the “Apply” button at the bottom of the dropdown menu
4. The system stores the new assignment

Alternative Flow:

- In 1, 2, 3: If A clicks beside the dropdown menu the menu closes again and the changes are lost.
- In 4: If there is an internal error a message is displayed to inform A

Remove User/Institute from a Category

ID: 1.17

Application: Access Category Manager

Primary Actor: User A

Description: The same way people are added to A's category it shall be possible to remove them again from a certain category
Priority: Medium (Phase 3)
Precondition: A has a user account and at least one access category defined. There is at least one other person already added to one of the categories
Trigger: A clicks on the small "x" button next to a user/institute
Flow:

1. The system removes the entry in the category for this person and updates the internal data

Alternative Flow:

- If there is an internal error a message is displayed to inform A

Remove an Access Category

ID: 1.18
Application: Access Category Manager
Primary Actor: User A
Description: It shall be possible to completely remove a category. If a access category gets removed, all access rights defined via this category shall be deleted as well
Priority: Low (Phase 3)
Precondition: A has a user account and at least one access category defined.
Trigger: A clicks the small "x" button next to the category
Flow:

1. The system removes the entire entry for the category and updates the internal data for all the containing users/institutes

Alternative Flow:

- If there is an internal error a message is displayed to inform A

Group records into folders

ID: 1.19
Application: User Record
Primary Actor: User A
Description: A shall have to possibility to combine certain records in a folder. This is very helpful e.g. in case of a certain injury that took several visits to a doctor, a surgery and physiotherapy to get healthy again. All the records in the folder are visible to all the people in the folders categories. It shall be possible to put the same record in multiple folders.
Priority: Low (Phase 3)
Precondition: A has a user account, at least one access category defined and at least two records.
Trigger: A clicks the "Create Folder" button in the User Record page.
Flow:

1. A dialog opens to ask for a folder name.
2. A enters a name for the folder.
3. A clicks "Create"
4. A new dialog is shown with a list of all records sorted descending by date
5. A clicks on the checkboxes next to all records he or she wants to group together
6. A clicks the "Save" button.
7. The system assigns all the selected records to the newly generated folder and continuous by showing the User Record page with a special view for folders.

Alternative Flow:

- In 1, 2, 3, 4, 5, 6: If A clicks the “Cancel” button the whole process is dismissed and no folder is created.
- In 4, 7: If A provided wrong input or if there was an external error a message is shown informing A about the problem.

Add an External App to User Account

ID: 1.20
Application: App Manager
Primary Actor: User A
Description: After an external app has been found A can add it to its user account and give it the right to upload data to his/her personal record. The app shall then be accessible via the app page
Priority: High (Phase 4)
Precondition: A has a user account and is logged in. There exist some live external applications in the system.
Trigger: A searches for applications and clicks the “Add Application” button next to one of them
Flow:

1. The system adds the application to the users list of applications
2. The system informs the application about the new user (This setup might be different for each application)

Alternative Flow:

- If there is an internal error a message is displayed to inform A

Remove an External App from User Account

ID: 1.21
Application: App Manager
Primary Actor: User A
Description: In the App Manager A shall be able to refuse access rights from a certain app and to remove the app from his/her app page. If need be A shall also be able to delete all the app data saved to his/her personal record
Priority: Low (Phase 4)
Precondition: A has a user account and is logged in the system. A has added at least one external application to his list of applications.
Trigger: A clicks the “Remove Application” button next to an application.
Flow:

1. A dialog is shown to ask the user if he or she really wants to delete this application.
2. A clicks the “Yes” button
3. Another dialog is shown to ask the user if he or she also wants to delete all the data produced by this application.
4. A clicks the “Yes” button once again.
5. The system removes the app from the users list of applications and removes all the associated data. The more the system indicates the application about the deletion (which might react differently depending on the individual application)

Alternative Flow:

- In 1, 2, 3: A clicks the “Cancel” button and the dialogs are dismissed without deleting anything.

- In 4: A clicks the “No” button and the system removes the app as defined in 5. but without deleting the associated data.
- In 5, alt 4: If there is an internal error a message is displayed to A to inform him/her about further steps to solve the issue.

Add Record from Ancestors to Personal Record

ID: 1.22
Application: User Record
Primary Actor: User A
Description: Give a doctor more information about possible hereditary diseases of A
Priority: Low (Phase 5)
Precondition: A and his mother and/or father or other ancestors have a system account
Trigger: Person A hits the button to add records of ancestors to his personal record on the right section of the Personal Record page
Flow:

1. A list of all members of the family access category gets displayed
2. Person A selects one or multiple of these contacts and presses the add ancestors data button
3. The selected ancestors of A get a notification to inform them about A’s request to access their data
4. The selected ancestors accept the request from A
5. An entry about the family members hereditary diseases is added to the personal record of person A and gets displayed in the respective section of his/her Personal Record page

Alternative Flow:

- In 2: Person A hits the “Cancel”-button and the popup dismisses
- In 4: The selected ancestor refuses to share his or her information and hits the “Refuse”-button. No entry is added to Person A’s personal record

Write other Users or Institutes a Message

ID: 1.23
Application: Messaging System
Primary Actor: User A
Description: A can send any other user a message by searching for the user and hitting the “Send Message” button.
Priority: Low (Phase 5)
Precondition: A and the user/institute to contact have both accounts in the system. A has searched for the other contact.
Trigger: Person A hits the button to send a message to the other contact.
Flow:

1. A dialog opens with a title and content textbox to enter text.
2. Person A enters the text to send and a title to the corresponding textboxes.
3. A hits the “Send” button
4. The system sends the message to the inbox of the addressed user or institute and informs the contact about the new message.
5. The recipient can open the received message in the Messaging System and read it.

Alternative Flow:

- In 2: Person A hits the “Cancel”-button and the dialog dismisses

- In 4: The sending could produce an error in which case a message is displayed to A, that the transmission of his or her message failed and he or she shall try again.

View Health Platform

ID: 1.24
Application: Health Platform
Primary Actor: User A
Description: As another part of the system there shall be a health platform with news, research highlights and annotations for surveys etc. Every user shall have access to this page
Priority: Optional (Phase 7)
Precondition: A has a user account and is logged in. There already exists a health platform and some Institutes or Research Accounts have provided any data.
Trigger: A clicks the "Health Platform" link in the navigation.
Flow:
The news page of the health platform is displayed. Any ongoing action is depending on the implementation of the platform.

Use Cases for Institutes

Create Profile

ID: 2.1
Application: Register
Primary Actor: Institute B
Description: Institutes shall be able to create pages about themselves and add information about the office such as opening hours, area of expertise etc.
Priority: High (Phase 1)
Precondition: The system is up and running
Trigger: B hits the "Register Institute" – button on the starting page
Flow:

1. A series of form elements is shown to B
2. B fills out the requested form elements
3. B clicks the "Register" button
4. The system stores the newly created account
5. B's browser navigates back to the login page

Alternative Flow:

- In 1, 2, 3: B clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to the welcome screen
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to B.

Change Profile Entry

ID: 2.2
Application: Profile Settings
Primary Actor: Institute B
Description: B shall be able to change the provided information anytime in the settings page

Priority: Medium (Phase 2)
Precondition: B has an account on the system and is logged in
Trigger: B hits the "Edit" button next to the profile information
Flow:

1. A series of form elements is shown to B filled out with the existing values.
2. B updates the requested form elements and adds possibly additional information.
3. B clicks the "Save" button
4. The system stores the data in the database.
5. B's browser navigates back to the profile page

Alternative Flow:

- In 1, 2, 3: B clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to profile page without any changes to the profile
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to B.

Delete Profile

ID: 2.3
Application: Profile Settings
Primary Actor: Institute B
Description: In the settings B shall be able to completely remove its account. All data which has been added to user records by this office shall remain
Priority: Low (Phase 2)
Precondition: B has an account and is logged into the system.
Trigger: B is on the Profile Settings page and clicks the "Delete Profile" button.
Flow:

1. A dialog is opened to ask B if he or she really wants to delete the profile
2. B clicks "Yes".
3. The system deletes all the entries for this account.
4. The browser navigates back to the welcome screen of the system.

Alternative Flow:

- In 1, 2: B clicks the "Cancel" button and the dialog is dismissed without any deletions.
- In 3: If there was an internal error, a message is shown to B to indicate it.

Add Consultation to Customers Record

ID: 2.4
Application: User Record
Primary Actor: Institute B
Description: After a customer A had a visit to B it shall be possible for B to add all information about the visit to A's personal record
Priority: High (Phase 2)
Precondition: A and B have accounts in the system. A has given B the rights to access its information.
Trigger: B clicks the "Add new Record" button next to the User Record of A
Flow:

1. B chooses what kind of record this shall be from a list of different options
2. A form with all possible information about the selected consultation is displayed
3. B fills out all the form fields

4. B clicks the “Add”-button
5. The system saves the new entry to A’s personal record
6. B is redirected to the User Record page where the new entry appears on the top

Alternative Flow:

- In 1: If B hits another button, the respective other form will be displayed. Compare with use cases for Users (1.4 and 1.7 - 1.9)
- In 5: If B’s input was illegal or some important information was missing as well as if there happened an error while saving an error message is displayed

Add Medicine or Prescription to Customer Record

ID: 2.5
Application: User Record
Primary Actor: Institute B
Description: See use case 1.9. B shall be able to add prescribed medicine to a customer’s personal record
Priority: High (Phase 2)
Precondition: A and B have accounts in the system. A has given B the rights to access its information.
Trigger: B clicks the “Add new Record” button next to the User Record of A
Flow:

1. B chooses the option to add a new drug from the list of different record options
2. See use case 2.4 starting at 2.

Alternative Flow: see use case 2.4

Add File to Customer Record

ID: 2.6
Application: User Record
Primary Actor: Institute B
Description: After a customer A had a visit to B it shall be possible for B to add files like MRIs or X-rays to A’s personal record
Priority: Medium (Phase 2)
Precondition: A and B have accounts in the system. A has given B the rights to access its information.
Trigger: B needs to add a file like a X-ray to a record
Flow:
This will just be another option in the entry of new Consultations as described in use case 2.4

Add Person/Institute to Customer Access Category

ID: 2.7
Application: Access Category Manager
Primary Actor: Institute B
Description: Similar to use case 1.15 B shall be able to create access categories and add users to a predefined customer category
Priority: High (Phase 3)
Precondition: An access category exists and B has an account on the system. There is a second person A in the system, a user, an institute or a research account
Trigger: B clicks on the category dropdown menu next to A
Flow:

1. On the dropdown list all access categories are listed next to a checkbox
2. B checks the desired checkboxes next to the categories
3. B clicks the "apply"-button at the end of the dropdown list
4. The system saves the changes

Alternative Flow:

- In 1,2: B clicks somewhere on the page next to the dropdown menu and the menu disappears
- In 2: B clicks the "Cancel"-button and the dropdown menu disappears
- In 4: If there happened an error while saving, a message gets displayed

Remove Person from Customer Access Category

ID: 2.8
Application: Access Category Manager
Primary Actor: Institute B
Description: The same way as B added users it shall be possible to remove them again from a access category
Priority: Low (Phase 3)
Precondition: B has a user account and at least one access category defined. There is at least one other person already added to one of the categories
Trigger: B clicks on the small "x" button next to a user/institute
Flow:

1. The system removes the entry in the category for this person and updates the internal data

Alternative Flow:

- If there is an internal error a message is displayed to inform B ***Contact Customers and Other Doctors via Messaging System***

ID: 2.9
Application: Messaging System
Primary Actor: Institute B
Description: B can send any user or other institute a message by searching for the user and hitting the "Send Message" button.
Priority: Low (Phase 5)
Precondition: B and the user/institute to contact both have accounts in the system. B has searched for the other contact.
Trigger: B hits the button to send a message to the other contact.
Flow:

1. A dialog opens with a title and content textbox to enter text.
2. B enters the text to send and a title to the corresponding textboxes.
3. B hits the "Send" button
4. The system sends the message to the inbox of the addressed user or institute and informs the contact about the new message.
5. The recipient can open the received message in the Messaging System and read it.

Alternative Flow:

- In 2: B hits the "Cancel"-button and the dialog dismisses
- In 4: The sending could produce an error in which case a message is displayed to B, that the transmission of his or her message failed and he or she shall try again.

Add Entries to the Health Platform Wiki

ID: 2.10
Application: Health Platform
Primary Actor: Institute B
Description: Institutes shall be allowed to create entries to the Health Platform Wiki which shall help customers to find important information about sicknesses etc. B shall be able to edit entries added by B itself as well as from others if need be. Every change to an entry shall be recorded similar to Wikipedia to keep the Wiki on a high standard
Priority: Optional (Phase 7)
Precondition: B has an account in the system and is logged in. B has navigated to the health platform.
Trigger: B has clicked the "Add entry" button next to existing entries.
Flow:

1. A series of form elements is shown to B for different input possibilities.
2. B fills out the requested form elements
3. B clicks the "Publish" button
4. The system stores the newly created entry in the database and publishes it for everyone to see.
5. B's browser navigates back to the overview page with all the entries including the new one

Alternative Flow:

- In 1, 2, 3: B clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to the overview page without changes to it
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to B.

Edit Entries in the Health Platform Wiki

ID: 2.11
Application: Health Platform
Primary Actor: Institute B
Description: Institutes shall be allowed to create entries to the Health Platform Wiki which shall help customers to find important information about sicknesses etc. B shall be able to edit entries added by B itself as well as from others if need be. Every change to an entry shall be recorded similar to Wikipedia to keep the Wiki on a high standard
Priority: Optional (Phase 7)
Precondition: B has an account in the system and is logged in. B has navigated to the health platform.
Trigger: B has clicked the "Edit" button next to existing entries.
Flow:

1. A series of form elements is shown to B for different input possibilities filled out with the existing values
2. B changes the requested form elements
3. B clicks the "Save" button
4. The system stores the edited entry in the database and updates it for everyone to see.
5. B's browser navigates back to the overview page with all the entries including the updated one

Alternative Flow:

- In 1, 2, 3: B clicks on the “Cancel” button, the form elements are dismissed and the browser navigates back to the overview page without changes to it.
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to B.

Use Cases for Research Accounts

Create Profile

ID: 3.1
Application: Register
Primary Actor: Research Account C
Description: C shall be able to create an account and a personal page describing what kind of company is behind and the topics they are doing research on
Priority: High (Phase 6)
Precondition: C is on the welcome screen of the system.
Trigger: C hits the “Register new Research Account” button
Flow:

6. A series of form elements is shown to C for different input possibilities.
7. C fills out the requested form elements
8. C clicks the “Register” button
9. The system stores the newly created account
10. C’s browser navigates back to the login page

Alternative Flow:

- In 1, 2, 3: C clicks on the “Cancel” button, the form elements are dismissed and the browser navigates back to the welcome screen
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to C.

Change Profile Entry

ID: 3.2
Application: Profile Settings
Primary Actor: Research Account C
Description: C shall be able to change the provided information in the settings
Priority: Low (Phase 6)
Precondition: C has registered an account and logged into the system.
Trigger: C clicks the “Edit” button next to the user profile
Flow:

6. A series of form elements is shown to C for different input possibilities filled out with the existing values.
7. C updates the requested form elements and adds possibly additional information.
8. C clicks the “Save” button
9. The system stores the data in the database.
10. C’s browser navigates back to the profile page

Alternative Flow:

- In 1, 2, 3: C clicks on the “Cancel” button, the form elements are dismissed and the browser navigates back to profile page without any changes to the profile
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to C.

Delete Profile

ID: 3.3
Application: Profile Settings
Primary Actor: Research Account C
Description: C shall be able to completely remove its account from the system
Priority: Low (Phase 6)
Precondition: C has an account and is logged into the system.
Trigger: C is on the Profile Settings page and clicks the “Delete Profile” button.
Flow:

5. A dialog is opened to ask C if he or she really wants to delete the profile
6. C clicks “Yes”.
7. The system deletes all the entries for this account.
8. The browser navigates back to the welcome screen of the system.

Alternative Flow:

- In 1, 2: C clicks the “Cancel” button and the dialog is dismissed without any deletions.
- In 3: If there was an internal error, a message is shown to C to indicate it.

Find People Query Engine

ID: 3.4
Application: Query Engine
Primary Actor: Research Account C
Description: If e.g. C wants to find people with illness X who have symptoms Y 1000 people who run more than 15km a week
Priority: Middle (Phase 6)
Precondition: C has a research account on our system and some users have allowed access to their records for research purposes
Trigger: C navigates to the Query Engine
Flow:

1. C fills out the presented form (Illness, symptoms, age, gender, sports activities, …) according to the documentation
2. C hits the “Find People”- button
3. While the systems tries to answer the query, a dialog is displayed to indicate work in progress
4. When the query is answered, the dialog is dismissed and a list of all the people found is shown with a preview of their records and a button to contact them. Besides the list there are statistics about how many people are found in total that match the query as well as a possibility to contact all or a certain amount of the people

Alternative Flow:

- In 1: C hits the info image next to each form field to get a dialog with a detailed description
- In 4: If there are no results to the query, C entered wrong input or if there was an error during execution of the query a message appears to inform C

Statistics Query Engine

ID: 3.5
Application: Query Engine
Primary Actor: Research Account C
Description: If e.g. C wants to find interesting facts about people with illness X or the percentage of people between 20 and 30 years of age who do more than 4 hours of sports a week
Priority: Low (Phase 6)
Precondition: C has a research account on our system and users have allowed access to their records for research purposes
Trigger: C navigates to the Query Engine
Flow:

1. C fills out the presented form (Illness, symptoms, age, gender, sports activities, …) according to the documentation
2. C hits the "Get Statistics"- button
3. While the systems tries to answer the query, a dialog is displayed to indicate work in progress
4. When the query is answered, the dialog is dismissed and the result of the query is displayed along with other useful information found by the system for similar queries

Alternative Flow:

- In 1: C hits the info image next to each form field to get a dialog with a detailed description
- In 4: If there are no results to the query, C entered wrong input or if there was an error during execution of the query a message appears to inform C

Contact User, Institutes and Research Accounts via Messaging System

ID: 3.6
Application: Messaging System
Primary Actor: Research Account C
Description: C can send any user, research account or institute a message by searching for it and hitting the "Send Message" button.
Priority: Low (Phase 6)
Precondition: C and the user/institute/research account to contact both have accounts in the system. C has searched for the other contact.
Trigger: C hits the button to send a message to the other contact.
Flow:

1. A dialog opens with a title and content textbox to enter text.
2. C enters the text to send and a title to the corresponding textboxes.
3. C hits the "Send" button
4. The system sends the message to the inbox of the addressed user, research account or institute and informs the contact about the new message.
5. The recipient can open the received message in the Messaging System and read it.

Alternative Flow:

- In 2: C hits the "Cancel"-button and the dialog dismisses
- In 4: The sending could produce an error in which case a message is displayed to C, that the transmission of his or her message failed and he or she shall try again.

Publish Entries on the Health Platform News Page

ID: 3.7
Application: Health Platform
Primary Actor: Research Account C
Description: Similar to use case 2.12 C shall be able to contribute to the Health Platform Wiki as well as add news about new research etc.
Priority: Optional (Phase 7)
Precondition: C has an account in the system and is logged in. C has navigated to the health platform.
Trigger: C has clicked the "Add entry" button next to existing entries.
Flow:

11. A series of form elements is shown to C for different input possibilities.
12. C fills out the requested form elements
13. C clicks the "Publish" button
14. The system stores the newly created entry in the database and publishes it for everyone to see.
15. C's browser navigates back to the overview page with all the entries including the new one

Alternative Flow:

- In 1, 2, 3: C clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to the overview page without changes to it
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to C.

Edit Entries on the Health Platform News Page

ID: 3.8
Application: Health Platform
Primary Actor: Research Account C
Description: Similar to use case 3.13 C shall be able to contribute to the Health Platform Wiki as well as add news about new research etc.
Priority: Optional (Phase 7)
Precondition: C has an account in the system and is logged in. C has navigated to the health platform.
Trigger: C has clicked the "Edit" button next to existing entries.
Flow:

6. A series of form elements is shown to C for different input possibilities filled out with the existing values
7. C changes the requested form elements
8. C clicks the "Save" button
9. The system stores the edited entry in the database and updates it for everyone to see.
10. C's browser navigates back to the overview page with all the entries including the updated one

Alternative Flow:

- In 1, 2, 3: C clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to the overview page without changes to it
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to C.

Create and Announce Studies and Surveys on the Health Platform or Info Page

ID: 3.9
Application: Health Platform
Primary Actor: Research Account C
Description: C may create surveys and announce studies visible for private people via the Health Platform or on the personal information page
Priority: Optional (Phase 6 and 7)
Precondition: C has an account in the system and is logged in. C has navigated to the health platform.
Trigger: C clicked the "Add new Survey" button.
Flow:

1. A series of form elements is shown to C for different input possibilities.
2. C fills out the requested form elements
3. C clicks the "Publish" button
4. The system stores the newly created survey in the database and publishes it for everyone to see.
5. C's browser navigates back to the overview page with all the entries including the new one

Alternative Flow:

- In 1, 2, 3: C clicks on the "Cancel" button, the form elements are dismissed and the browser navigates back to the overview page without changes to it
- In 4: If the provided input is wrong or there was an internal error a message is displayed to indicate the problem to C.

Use Cases for External Apps and App Developers

Create Application Using our API

ID: 4.1
Application: External IDE
Primary Actor: App Developer D
Description: According to our API description a developer shall be able to create an application which can be included to our system in a similar way as Facebook does for games etc.
Priority: High (Phase 4)
Precondition: The API for our system exists and there is documentation about it. Plus the developer has basic knowledge in the languages used to write the system.
Trigger: A good idea by the developer.
Flow: There is no interaction with our system at that point besides reading the documentation

Create Profile

ID: 4.2
Application: Register
Primary Actor: App Developer D
Description: D shall be able to register an account for the application he or she is building that contains information about the application and helps the users to find and install it. The more it tells the users what kind of information this application needs from them.

Priority: High (Phase 4)
Precondition: The system is up and running
Trigger: D hits the "Register Application" button on the welcome page of the system.
Flow:

1. D fills out the form displayed with all the information needed
2. D hits the "Register" button
3. The system stores the data and navigates to the login screen
4. D can now login with the provided data

Alternative Flow:

- In 1: D can hit the "Cancel" button anytime to stop the registration process
- In 3: If D provided wrong or missing input or if there was an error while storing the data, D gets informed accordingly.

Integrate App in System and Make it Visible

ID: 4.3
Application: App Manager
Primary Actor: App Developer D
Description: In the App Manager D shall be able to upload to and make a finished app live in our system and to make it publicly accessible for users.
Priority: High (Phase 4)
Precondition: D has registered the application in the system and provided the necessary description
Trigger: D clicks the "Upload App" button on the App Manger
Flow:

1. A dialog with a file selection will be shown.
2. D selects the right file(s) and clicks the "Upload" button
3. A dialog is opened to ask D to agree to the general terms and conditions
4. D sets the according checkmark and hits the "Proceed" button
5. Another dialog indicates D that the system is now trying to integrate the new application and that he or she will be informed, when the application is live.
6. The system makes a series of automatic background checks on the application and tries to integrate it.
7. When all the checks are passed and the integration finished a message is sent to the D, that the application is now live.

Alternative Flow:

- In 1, 2, 3, 4: D hits the "Cancel" button and the upload process is dismissed
- In 5, 6: If there was an error reading the uploaded file or one of the checks did not go well or the integration was erroneous a message will be send to D that indicates more detail about the problem.

Change Profile Entry

ID: 4.4
Application: Profile Settings
Primary Actor: App Developer D
Description: D shall be able to change to provided information about the application in the settings
Priority: Low (Phase 4)
Precondition: D has registered the application

Trigger: D clicks the "Edit" button next to the app information
Flow:

1. D changes the entries in the displayed form elements
2. D clicks the "Save" button
3. The system saves the changes and displays a success message

Alternative Flow:

- In 1, 2: D clicks the "Cancel" button and the editing process is dismissed
- In 3: If the provided data was wrong or there was an internal error, e message indicating the problem will be shown to D

Update App

ID: 4.5
Application: App Manager
Primary Actor: App Developer D
Description: In the App Manager D shall be able to update its apps with newer versions
Priority: Medium (Phase 4)
Precondition: D has made an account for the application and uploaded the application. The application integration into the system was successful and the application is now live
Trigger: D hits the "Update" button in the App Manager
Flow:
Same flow as in use case 4.3

Delete App

ID: 4.6
Application: App Manager
Primary Actor: App Developer D
Description: If an app causes problems or any other kind of reasons it shall be able for D to remove its apps from the system. Uploaded data shall remain on the system if not deleted by users themselves
Priority: Low (Phase 4)
Precondition: D has made an account for the application and uploaded the application. The application integration into the system was successful and the application is now live
Trigger: D hits the "Delete App" button in the App Manager
Flow:

1. A dialog asks D if he really wants to delete the application and warns him that this process is not irreversible.
2. D clicks the "Delete Anyway" button
3. The system deletes the application while not deleting the data in the users accounts
4. The system informs all the users of the application about the deletion of it and asks them if they want to keep the data or delete it.

Alternative Flow:

- In 1, 2: D clicks the "Cancel" button and the dialog is dismissed without deletion of the application
- In 3, 4: If there is an internal error while deleting the application a message to the system admin will be sent and D gets informed, that we try to solve the problem as soon as possible.

Delete Profile

ID: 4.7
Application: Profile Settings
Primary Actor: App Developer D
Description: D shall be able to completely remove its account from the system with all of its apps
Priority: Low (Phase 4)
Precondition: D has an account for the application
Trigger: D hits the "Delete Profile" button
Flow:

1. A dialog asks D if he really wants to delete the account including the application (if there still is one).
2. D clicks the "Delete" button to accept
3. The system deletes the application (if not done before). See use case 4.6.
4. The system deletes all the profile information for the application and navigates back to the welcome screen of the system.

Alternative Flow:

- In 1, 2: If D clicks the "Cancel" button the dialog is dismissed and nothing gets deleted.
- In 3, 4: If there is an internal error while deleting the application a message to the system admin will be sent and D gets informed, that we try to solve the problem as soon as possible.

Save Data to Users Record

ID: 4.8
Application: External Application
Primary Actor: External Application
Description: External applications shall be able to upload data collected about a user A to A's personal record provided A did allow the app to do so
Priority: High (Phase 4)
Precondition: A has a user account. There is an account for the external application and the application is accepted in the system and live.
Trigger: The external application calls the according API function
Flow:
This is depending on the actual data and the design of the API. In general the external application calls the API according to the rules defined in the documentation. The system executes the desired function and sends back an answer.
Alternative Flow:
If there happens to be an error in the process of answering the API call an error message will be sent back to the caller.

Use User Record Data

ID: 4.9
Application: External Application
Primary Actor: External Application
Description: If a user A allows the external application to access its personal record it shall be able to use this data for any kind of calculations to improve the user experience
Priority: Optional (Phase 4)
Precondition: A has an account on the system and granted the external application access to its data. The external application is registered on the page and live.

Trigger:
The external application calls the according API function.
Flow:
This is depending on the actual data and the design of the API. In general the external application calls the API according to the rules defined in the documentation. The system executes the desired function and sends back an answer.
Alternative Flow:
If there happens to be an error in the process of answering the API call an error message will be sent back to the caller.

Additional Use Cases

Become an Organ Donor

ID: 5.1
Application: Personal Record
Primary Actor: Private person A
Description: Add/Create donation card
Priority: Optional
Precondition: Person A has a system account
Trigger: Person A hits the button to add organ donation card to his/her personal record on the right section of the Personal Record page
Flow:

1. A popup appears to let the user either select adding a new card or importing an existing one from Swisstransplant
2. Person A selects the "Add new Card" - button
3. Another popup is displayed with a form to decide what to do with the persons organs in case of death
4. Person A fills out the form and hits the "Create Card" – button
5. A confirm message appears who asks the user if this is his final decision
6. Person A hits the "OK"-button
7. The system send the information to Swisstransplant and displays a message that the request was sent to Swisstransplant
8. When the confirmation from Swisstransplant arrives, the data is saved to Person A's personal record. On the Personal Record page of Person A a new entry for the donation card is displayed

Alternative Flow:

- In 2: Person A hits the "Add existing Card"-button. Then a login form for Swisstransplant is displayed that lets the user add an existing Card. When user logins in, continue with 7.
- In 4: Person A hits the "Cancel"- button and the form popup dismisses
- In 6: Person A hits the "Cancel"- button and the confirmation message dismisses. The form is shown again
- In 8: When there is no confirmation message from Swisstransplant after a certain timeout or an error happened, the user gets a new notification

Find Organ Donors

ID: 5.2
Application: Research Find People Query Engine
Primary Actor: Swisstransplant
Description: Find people who are willing to donate organs
Priority: Optional
Precondition: Swisstransplant has a system account
Trigger: Employee of Swisstransplant navigates to Research Find People Query Engine
Flow: Same as use case 1.2 where form is filled out in a way that it queries for people who have a organ donation card
Alternative Flow:

- See use case 1.2

Coordinate Calendars

ID: 5.3
Application: Doctor's Office Record
Primary Actor: Employee of doctor's office
Description: Share appointments book with customers to improve schedule
Priority: Optional
Precondition: Doctor's office has a system account
Trigger: Employee of doctor's hits "Share Calendar"- button on the Doctor's Office Record page
Flow:

1. On a popup the user gets a selection of different calendars like Gmail, Outlook, iCal, etc.
2. Employee clicks the calendar he/she wants to connect
3. On a second view employee enters the login data for the calendar
4. A message indicates login success and that the synchronization will take part in the background
5. System synchronizes with the external calendar and adds a calendar view to the Doctor's Office Record

Alternative Flow:

- In 1 and 3: Employee clicks the "Cancel"-button and the popup dismisses
- In 4: If there was a login or connection error a message is displayed
- In 5: If synchronization resulted in an error. A notification is sent to the Doctor's Office account to inform about the synchronization error

Doctor Rating System

ID: 5.4
Application: Doctor's Office Record
Primary Actor: Private person A
Description: Rate doctors / hospitals etc.
Priority: Optional
Precondition: Doctor's Office Record exists and Person A has a system account
Trigger: Person A navigates to Doctor's Office Record B
Flow:

1. Next to the office description an indicator with five stars tells the user about the rating given to the office by other customers

2. Person A clicks on the five stars
3. On a popup dialog Person A can give a rating on its own by clicking on one of the five stars
4. The system adds the new rating to the office rating record and updates the average value which is indicated by the five stars as described on 1.

Alternative Flow:

- In 1: Next to the five stars there is also a small selection of the latest 3 comments given to the office. If Person A clicks on them, a complete list of all the comments will be shown with the possibility to add one on its own or to edit the own comment
- In 3: If Person A has already rated the office, the value entered the last time is shown and updated if user changes the value

Dealing with Prescriptions

ID: 5.5
Application: Personal Record
Primary Actor: Private Person A, Doctor's Office B
Description: To make prescriptions easier to deal with, doctors can issue them directly on the customers personal record from where they are accessible paperless from all over the world
Priority: Low
Precondition: There are users and doctors in the system. A has added B to his circles
Trigger: B is logged in and navigates to the personal record of A
Flow:

1. B hits the "Add Prescription"-button on A's personal record page
2. In the form that appears B enters all the required data and hits the "Add"-button
3. The system adds the prescription data to the personal record of A. It is now accessible for A via his/her Personal Record page

Alternative Flow:

- In 2: B or C hits the "Dismiss"-button and the data gets deleted
- In 3: If there was an error or B entered wrong input a message appears to inform