*BWF (Browser Work Flow)*

Software Requirements Specification

Table of Contents

[Figures 4](#_Toc457201315)

[Document History 5](#_Toc457201316)

[Introduction 5](#_Toc457201317)

[Purpose 5](#_Toc457201318)

[Scope 5](#_Toc457201319)

[Overall description 6](#_Toc457201320)

[Product perspective 6](#_Toc457201321)

[Product functions 6](#_Toc457201322)

[User characteristics 6](#_Toc457201323)

[Requirements 7](#_Toc457201324)

[Functional requirements 7](#_Toc457201325)

[General 7](#_Toc457201326)

[Project 7](#_Toc457201327)

[UI Requirements 10](#_Toc457201328)

[General 10](#_Toc457201329)

[Application 11](#_Toc457201330)

[Project 13](#_Toc457201331)

[File 17](#_Toc457201332)

[Nonfunctional Requirements 20](#_Toc457201333)

[Entities 20](#_Toc457201334)

[User 20](#_Toc457201335)

[Project 20](#_Toc457201336)

[Activity 20](#_Toc457201337)

[File 20](#_Toc457201338)

[Assumption 21](#_Toc457201339)

[Metadata 21](#_Toc457201340)

[Decision 21](#_Toc457201341)

[Use Cases 21](#_Toc457201342)

[Actors / User roles 21](#_Toc457201343)

[Scientist 22](#_Toc457201344)

[Manager 22](#_Toc457201345)

[Reviewer 22](#_Toc457201346)

[UC1 – Login 22](#_Toc457201347)

[UC2 – Create a project 22](#_Toc457201348)

[UC3– Show own projects 22](#_Toc457201349)

[UC4 – Edit a project 23](#_Toc457201350)

[UC5 – Show projects by user 23](#_Toc457201351)

[UC6 – Show projects to review 24](#_Toc457201352)

[UC7 – Open Project 24](#_Toc457201353)

[UC8 – Create Assumption 24](#_Toc457201354)

[UC9 – Modify Assumption 25](#_Toc457201355)

[UC10 – Invalidate Assumption 25](#_Toc457201356)

[UC11– Create Decision 25](#_Toc457201357)

[UC12– Modify Decision 26](#_Toc457201358)

[UC13 – Invalidate Decision 26](#_Toc457201359)

[UC14 – Show file content 26](#_Toc457201360)

[UC15 – Tag a Model 27](#_Toc457201361)

[UC16 – Remove tag from a model 27](#_Toc457201362)

[UC17 – Set QC Flag 27](#_Toc457201363)

[UC18 – Export Project 27](#_Toc457201364)

[UC19 – Copy Table 28](#_Toc457201365)

[UC20 – Audit Trail 28](#_Toc457201366)

[UC21 – Run record 28](#_Toc457201367)

[UC22– QC checklist 29](#_Toc457201368)

[UC23 – Assumptions table 29](#_Toc457201369)

[UC24 – Assign or change of reviewer(s)? 29](#_Toc457201370)

[UC25 – Assign or Update a reviewer 30](#_Toc457201371)

[UC26– Decision table 30](#_Toc457201372)

# Figures

[Figure 1 D3JS Flower chart at http://www.redotheweb.com/CodeFlower 9](#_Toc457201373)

[Figure 2 Sankey diagram example at https://bost.ocks.org/mike/sankey/ 10](#_Toc457201374)

[Figure 3 Login Screen 11](#_Toc457201375)

[Figure 4 Navigation 11](#_Toc457201376)

[Figure 5 Project list 13](#_Toc457201377)

[Figure 6 Create Project 13](#_Toc457201378)

[Figure 7 Project detail page 14](#_Toc457201379)

[Figure 8 QC Checklist 14](#_Toc457201380)

[Figure 9 Assumption list 15](#_Toc457201381)

[Figure 10 Decision list 15](#_Toc457201382)

[Figure 11 Run Record 16](#_Toc457201383)

[Figure 12 Assign User 16](#_Toc457201384)

[Figure 13 File Assumptions 18](#_Toc457201385)

[Figure 14 File Decisions 19](#_Toc457201386)

# Document History

|  |  |  |
| --- | --- | --- |
| Description | Author | Date |
| Initial Document | Christoph Stocker | 22.6.2016 |
| Added Requirements and Uses Cases | Christian Flandorfer | 26.6.2016 |
| *Extend* Requirements and Uses Cases | Christoph Stocker | 28.6.2016 |
| *Entities, Questions* | Christoph Stocker | 30.6.2016 |
| *Adapt document based on our online meeting 1.7.2016* | Christoph Stocker | 4.7.2016 |

# Introduction

This section gives a scope description and overview of everything included in this SRS document. In addition, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Purpose

The purpose of this document is to give a detailed description of all needed requirements for the DDMoRe system. It will explain system constraints, interface and interactions with other external systems. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## Scope

The goal of the project is to further develop the tool for tracking and reporting the provenance of activities, entities and decisions making up a pharmacometric data analysis project, with a view to facilitating updates of aspects of the analysis, or the entire analysis, after changes to inputs, and to facilitating reporting: audit logs, QC checklists and run records are three central examples.

Main features for a minimum deliverable

* Track entities, activities, decisions within a project, and how they relate to one another
* Basic visualizations of these relationships
* Extraction of information for reporting

References to the part that scinteco will provide is specified in Project Components as “BWF” (Browser Work Flow)

# Overall description

This section will give an overview of the whole integration. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## Product perspective

The BWF system is planned as a web application accessing the following provided services:

1. REST data service to get all the relevant information – e.g., projects, users, …
2. Authentication and Authorization service to do login and query user roles.

## Product functions

The product provides:

* Project page that shows files, metadata and recent activities on the project
* The user can add/ modify metadata (e.g. Tags, QC)
* The user can add/ modify assumptions
* The user can add/ modify decisions
* The user can export data.

## User characteristics

For the whole document, a User is defined as a Scientist or a Manager. A Manger could also be a Scientist.

#### Scientist

A Scientist can access projects, assigned files and perform activities on projects and the assigned files.

#### Manager

A Manager is responsible for a team of Scientists and can access their work (projects) readonly.

#### Reviewer

Reviewer is not a user role. The classification as Reviewer depends on the project. A Reviewer for a project is assigned by its owners.

# Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

## Functional requirements

This section includes the requirements that specify the actions added to the software system.

### General

#### FR1001 – Login

The user can login into the application by providing username and password. The user is assigned to one of the following roles:

* Scientist: A user can create or edit projects.
* Manager: A manager has one to many users. A manager has automatically read access to the projects of the assigned users. In addition, the manager can use the same functionality as a scientist.

#### FR1002 – Copy Tables

The user can copy the content of a table and paste it into a Word File. On copy, the content of a table is available in the clipboard, so the user can paste it into Word or whatever. A user can select two different content types to copy into clipboard:

1. Text (CSV format – semicolon as delimiter)
2. HTML

#### FR1003 – Show projects by owner

A user can show a list of the projects where he is the owner. A scientist can access only his own projects and a Manager can access the projects of all assigned users he is assigned to.

#### FR1004 – Show projects by Reviewer?

A user can show a list of projects where he is assigned as Reviewer.

#### FR1005 – Multi-Language support (i18n )

The application provides the language English via a property file and is therefore designed to easy translate this file into other languages for multi-language support.

### Project

#### FR2001 – Show Project parts as tables

The User can open a project. The Project overview shows Metadata:

* Project Owner
* Reviewer
* Manager
* An activity stream for the project

The project overview also shows the assigned files and their type which are:

* Input files
* Output files
* Assumptions
* Decisions

#### FR2002 – Assign Reviewer

The Scientist can assign reviewers to his project. A Reviewer can be any User.

#### FR2003 – Remove Reviewer

The Scientist can remove a reviewer from his project.

#### FR2004 – Add Assumption to a model

The Scientist can add one to many assumptions to a model.

An Assumption consists of:

* Type (Pharmacological, Physiological, Disease, Data, Mathematical/ Statistical)
* Justification
* Assumption Body
* Established
* Testable
* TestApproach
* TestOutcome

#### FR2005 – Modify Assumtion

The Scientist can modify an existing Assumption.

#### FR2006 – Remove Assumption

The Scientist can mark an existing Assumption with a flag to hide this Assumption in project.

#### FR2007 – Add Decision to an Entity

The Scientist can add one to many decisions to a project artifact. A Decision contains only a description of the decision.

#### FR2008 – Show graphical overview of a project

The user can switch to a graphical overview of a project. The visualization contains the project artefacts, users and software, and the actions they took. The chart functionality we provide is given and limited by the implementation of D3JS.

The interactive chart should be a D3JS flower chart: <http://www.redotheweb.com/CodeFlower>

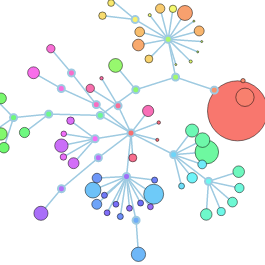


Figure 1 D3JS Flower chart at http://www.redotheweb.com/CodeFlower

#### FR2009 – Show file content (readonly)

The user can open a file and show the file content readonly. The file content is shown as text in the browser to avoid any modification and is available directly from GIT by its URL.

#### FR2010 – Tag Model

The Scientist can tag a model:

* Base Model
* Final Model
* Pivotal Model

#### FR2011 – Remove Tag

The Scientist can remove an existing tag.

#### FR2012 – Set QC Status

The Reviewer can set the QC Status (true = passed QC/ false = failed QC). In case of false, this should show all created project artifact where the project artifact marked as failed is the creator also as invalid. The QC Status is a special Tag.

#### FR2013 – Export Project

The Scientist can export the project and the Metadata and the Files that are assigned to a project. The export is a ZIP file containing the following files:

1. A XML file containing all project data description (e.g. input files, output files, assumptions …) - The XML structure must be defined.
2. All related files to the project.

#### FR2014 – Audit Trail

The Scientist can create the audit trail for a selected file. The Audit Trail contains all Entities, Agents, Activities and their relations. A filter will be provided to select relevant data show in the chart.

An Audit Trail is shown with PROV-O-Vis: <https://github.com/Data2Semantics/provoviz>

With PROV-O-Viz you can visualize any provenance graph expressed using the W3C PROV-O vocabulary as an intuitive Sankey diagram.

We build the Sankey diagram with the D3JS Sankey plugin: <https://bost.ocks.org/mike/sankey>

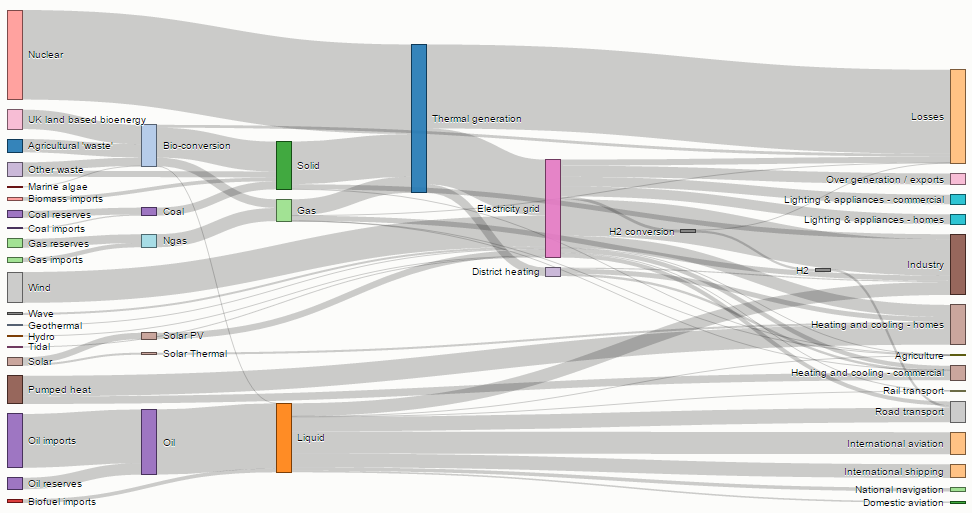


Figure 2 Sankey diagram example at https://bost.ocks.org/mike/sankey/

## UI Requirements

This section contains UI requirements specifying all needed user interfaces for the software system.

### General

#### UIR1001 – Project Focused UI

The UI is project focused.

#### UIR1002- Clean and Functional

The UI should show the information in a clean and functional way.

#### UIR1003- Look nice on mobile devices

The UI should also work and look nice on mobile devices but we don’t use a “Mobile First” approach for this project.

#### UIR1004- Menu

The menu should be collapsible to save horizontal space.

#### UIR1005- Icons

Use Font Awesome icons.

#### UIR1005- Tables

Tables should be:

1. Sortable
2. Filterable
3. Pageable
4. Exportable (Content is available from clipboard after export)

### Application

#### UIR2001 - Login

The user has to login to the system by providing username and password

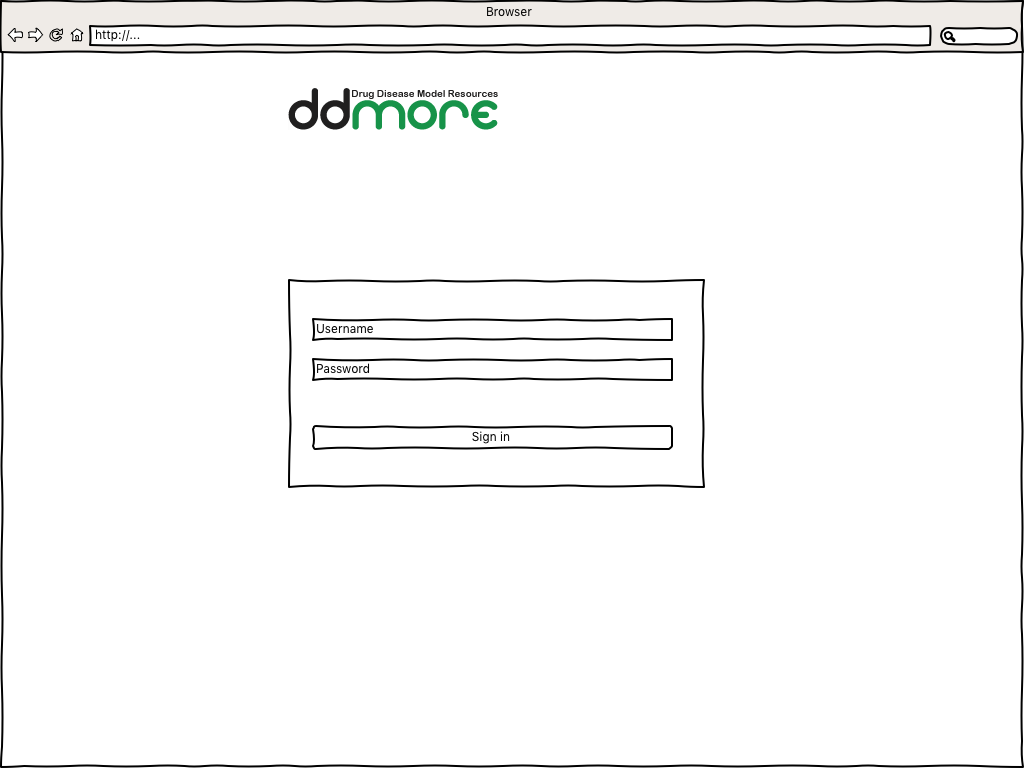


Figure 3 Login Screen

#### UIR2002 – Navigation

A Navigation Bar on the left side allows the user to navigate quickly between different areas.

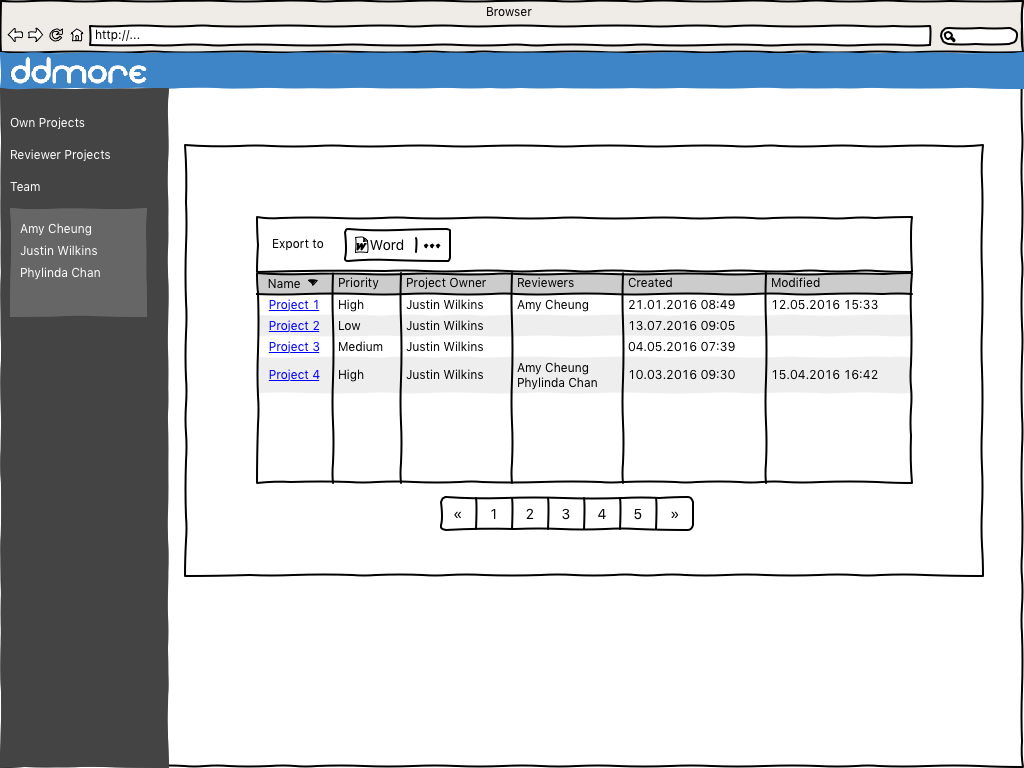


Figure 4 Navigation

### Project

#### UIR3001 – Project list

The project list gives an overview of the projects assigned to a specific user. The list is accessible via “Own Projects”, “Reviewer Projects” or the name of a team member.

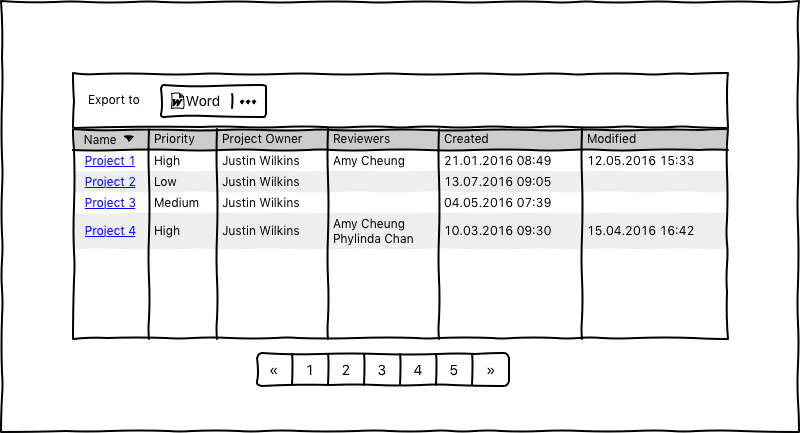


Figure 5 Project list

#### UIR3002 – Create Project

This page is needed to create a new project. The user has the option to provide a URL to an existing project repository or to create a new repository.

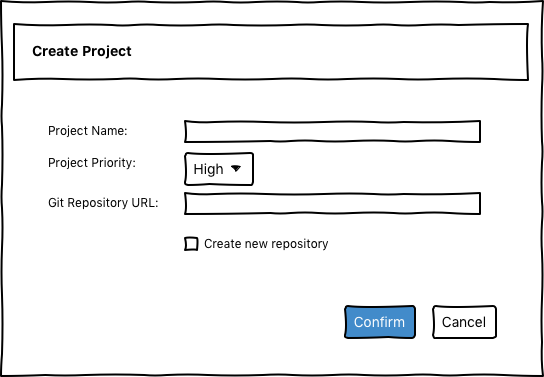


Figure 6 Create Project

#### UIR3003 – Project detail page

The detail page shows the associated files and meta data to the selected project.

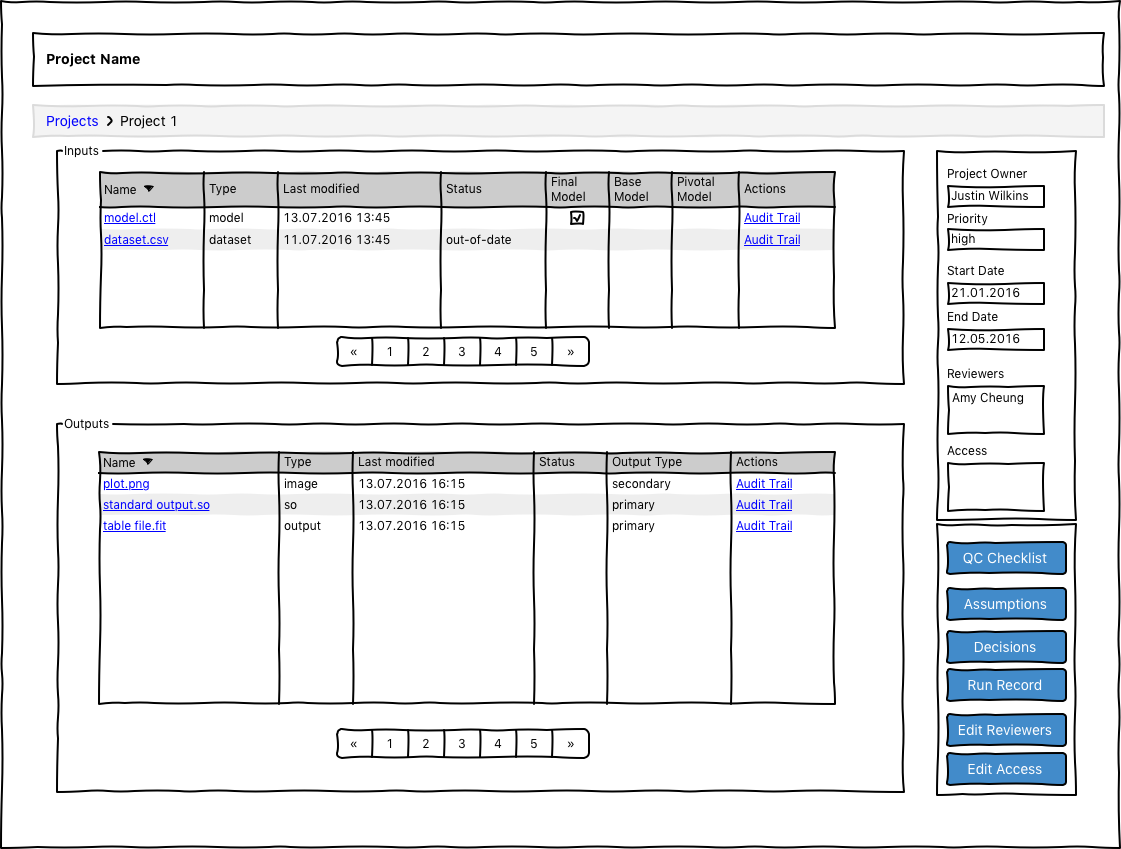


Figure 7 Project detail page

#### UIR3004 – QC Checklist

The user can create a QC checklist that for a project

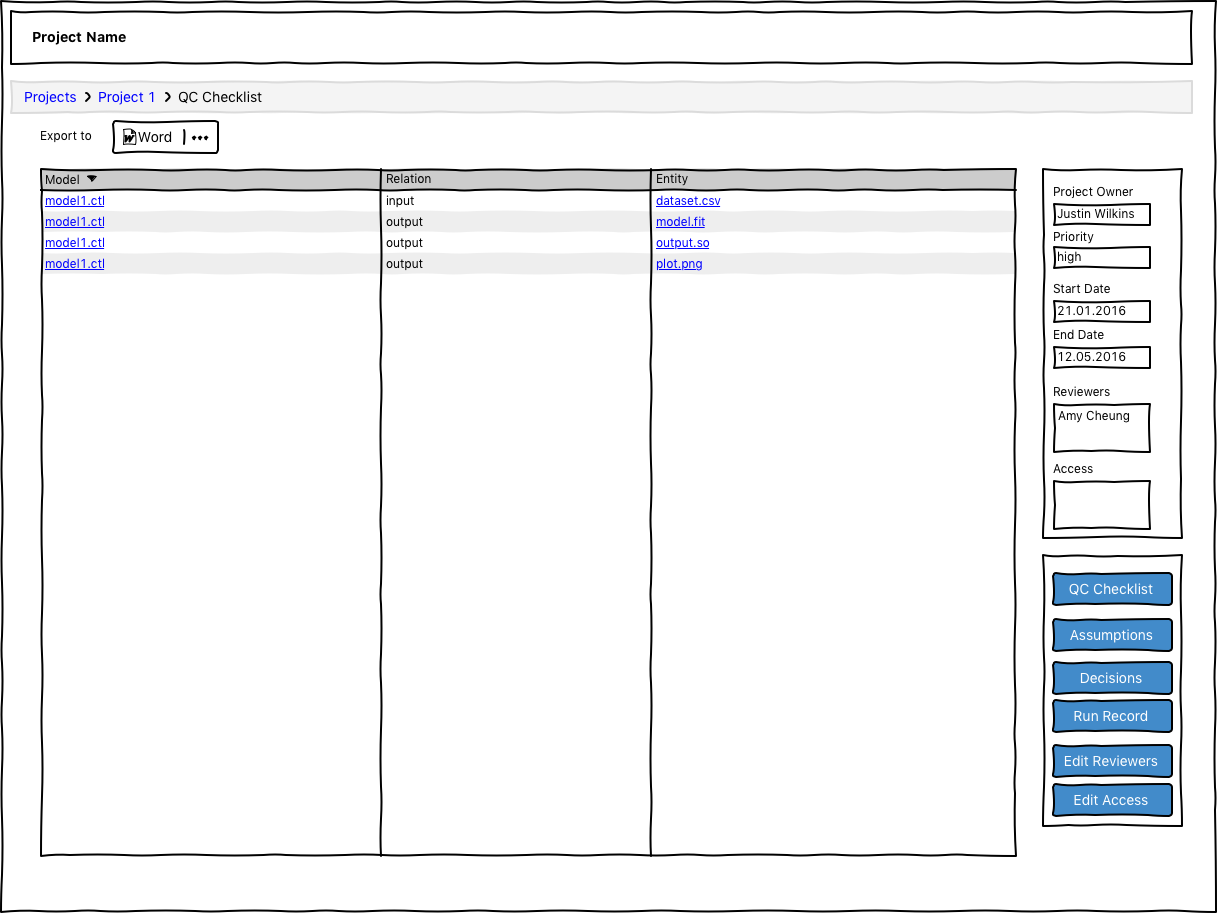


Figure 8 QC Checklist

#### UIR3005 – Assumptions

The user can create a list of assumptions based on all key steps in the current project

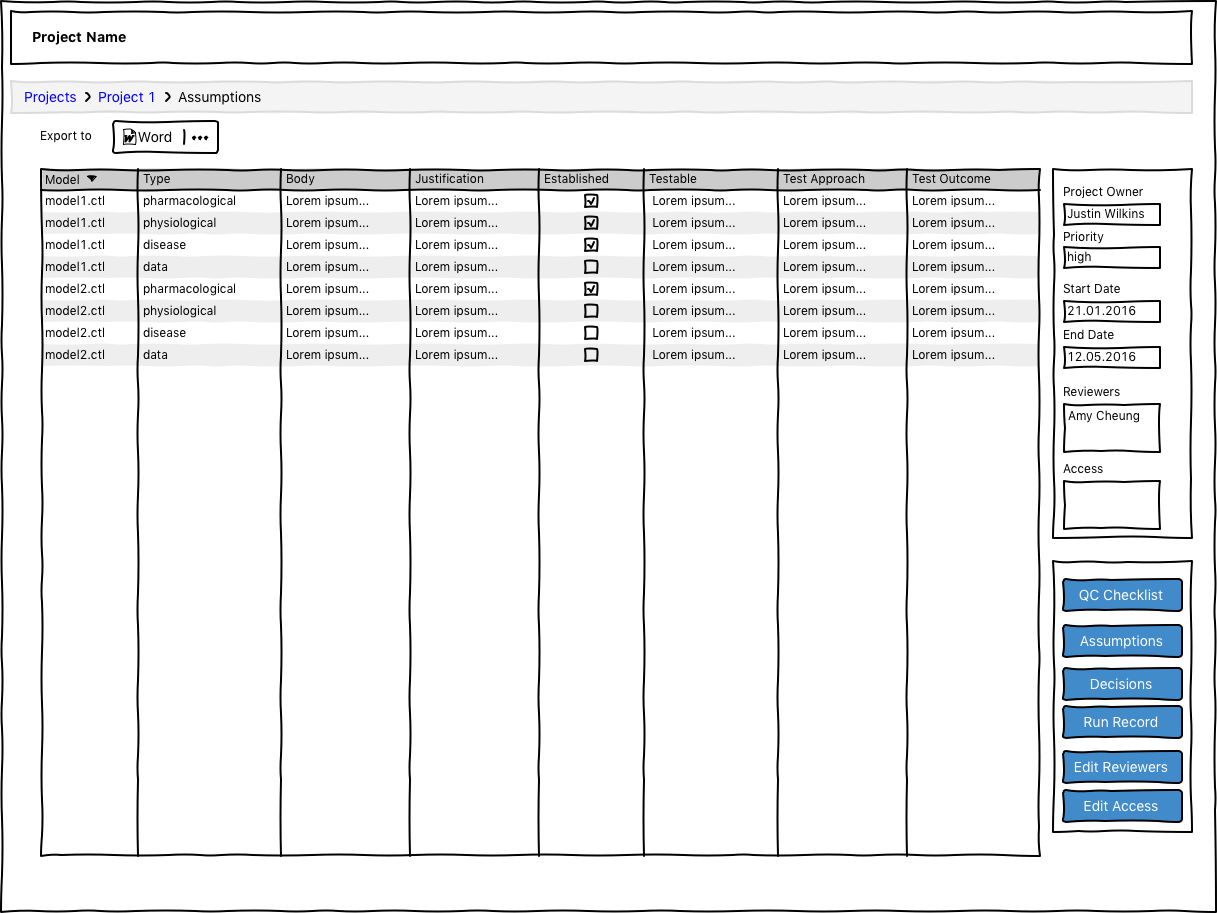


Figure 9 Assumption list

#### UIR3006 – Decisions

The user can create a list of decisions based on all key steps in the current project

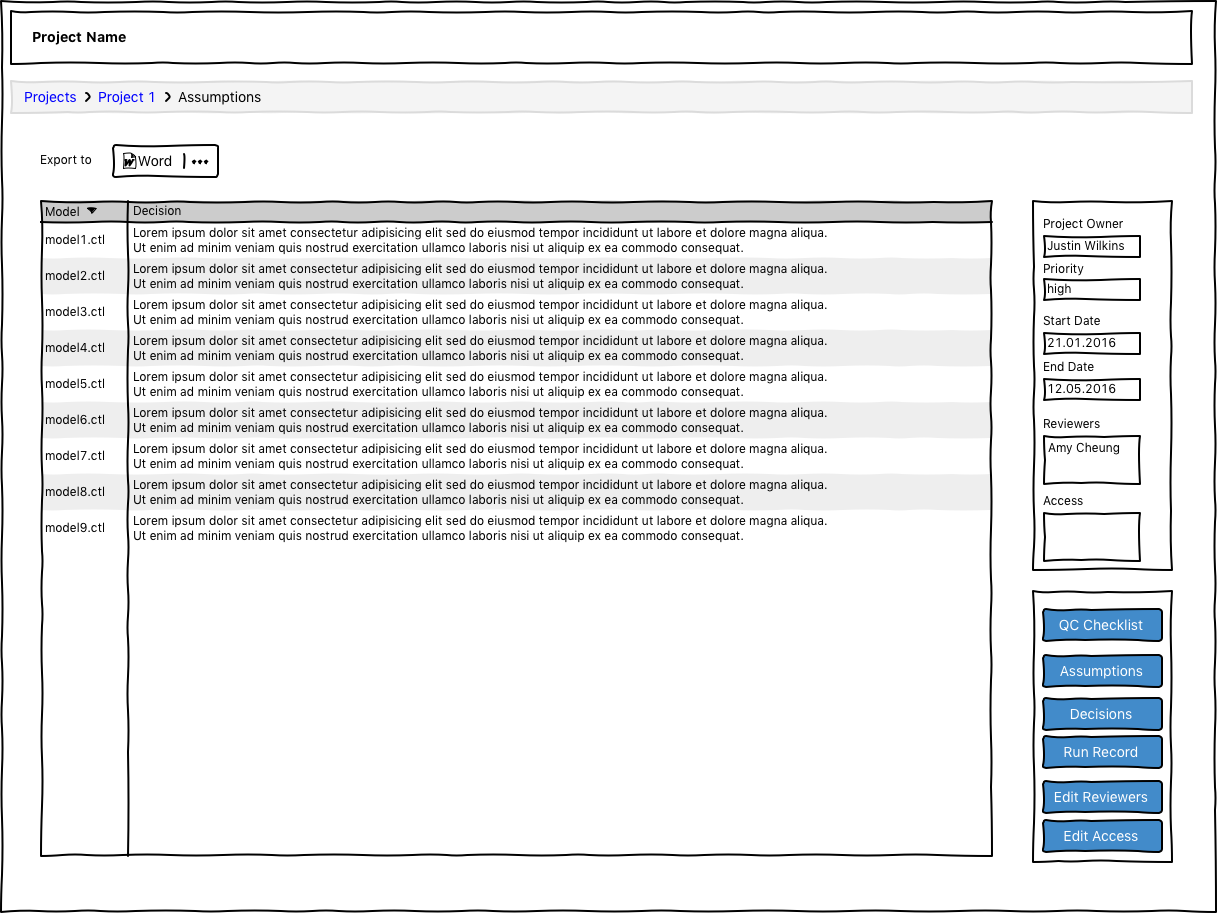


Figure 10 Decision list

#### UIR3007 – Run Record

The run record shows details like OFV, deltaOFV,… on the key steps of the current project.

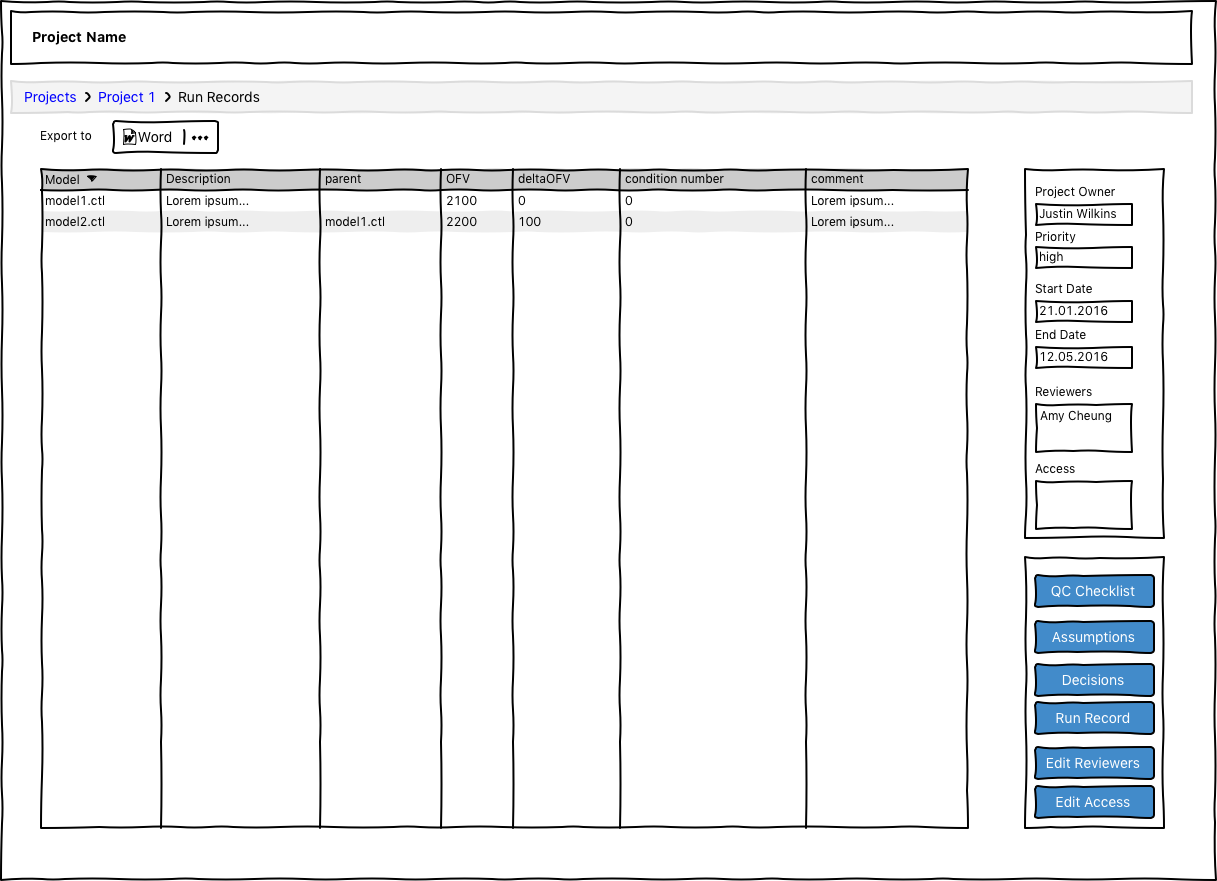


Figure 11 Run Record

#### UIR3006 – Assign Users

The owner of the project can assign users to the project by giving them access or assigning them as reviewer

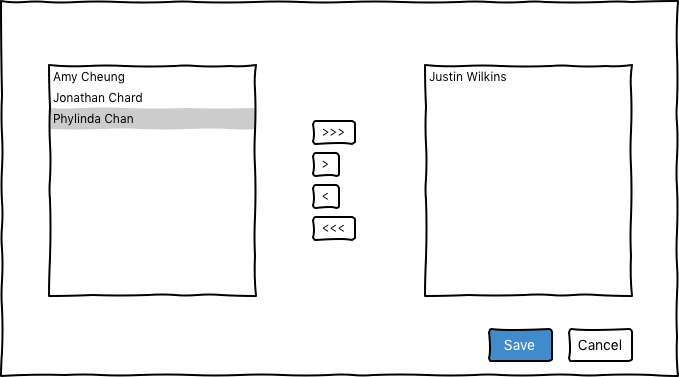


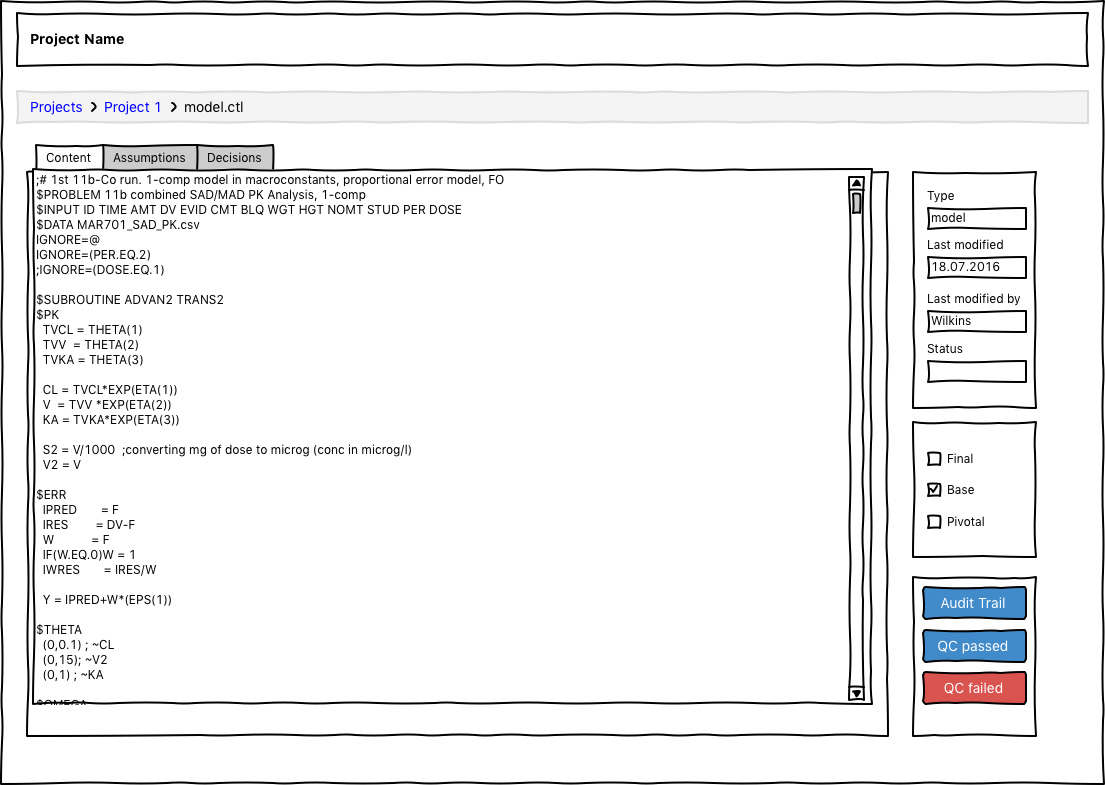
Figure 12 Assign User

### File

#### UIR4001 – File Content

The user can access the file content by navigating to a specific file. The file content is readonly. In addition the page shows detail information. Some parts are dependent on the file type.

* Model 🡪Show tags (final, base and pivotal)
* Output 🡪Type of output (primary, secondary)



#### UIR4002 – File Assumptions

This page is needed to display and edit the assumptions that are associated to a file

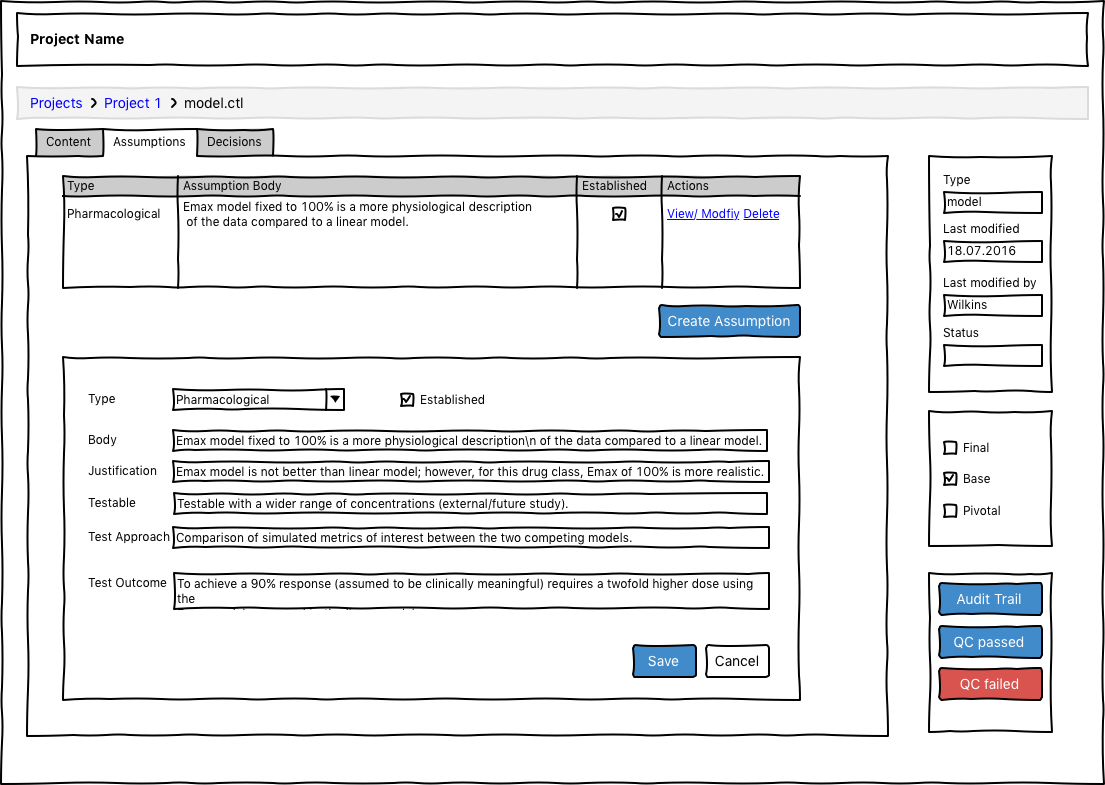


Figure 13 File Assumptions

#### UIR4003 – File Decisions

This page is needed to display and edit the assumptions that are associated to a file

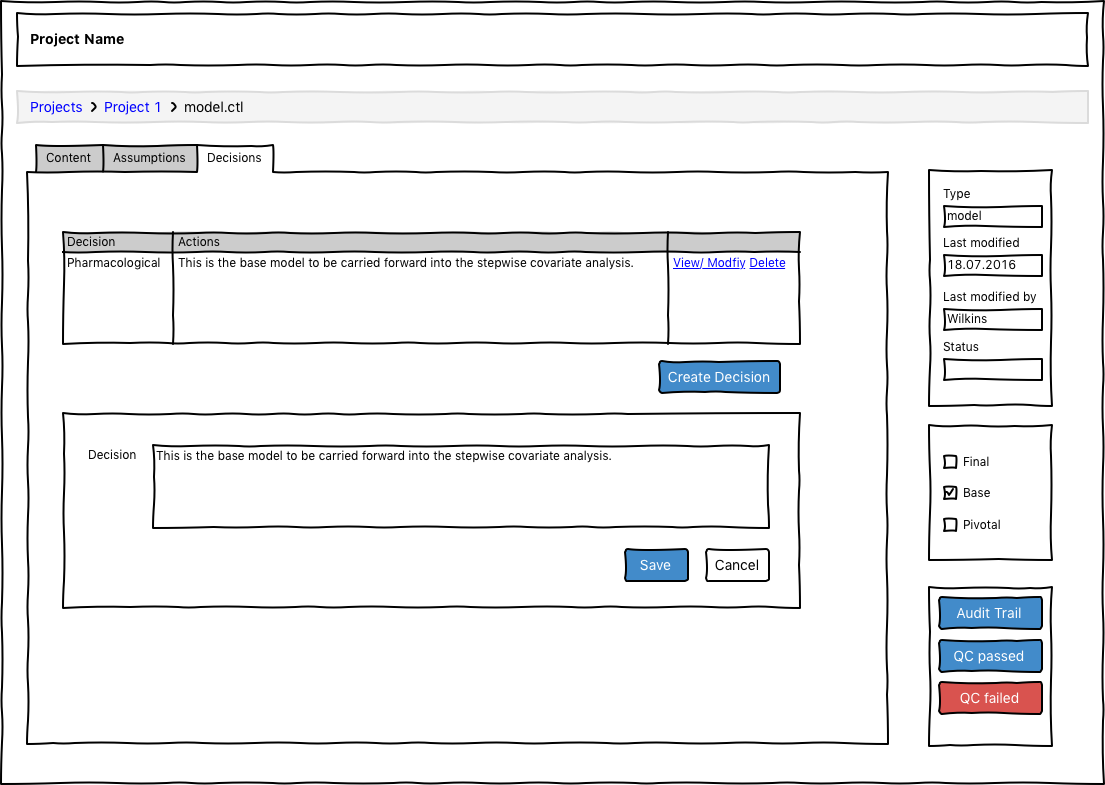


Figure 14 File Decisions

## Nonfunctional Requirements

This section contains non-functional requirements needed for the software system.

#### NFR1001 – Local Deployment

The application should run on a local PC without any deployment. A BAT or SH file (depending on the operating system of the local PC) should be provided to start the web application.

#### NFR1002 – Privileges

Each user is assigned a specific role represented in the system. With an assigned role the user acquires the role-specific functionality in the software system.

#### NFR1003 – Data protection

All information that is subject to data protection rules needs to be secured in a proper way.

# Entities

### User

* Username
* First name
* Last name
* Email
* Company: *optional*
* Location: *optional*
* User Type (Scientist, Manager)
* Projects

### Project

* Id
* Name
* Steps
* Users (Owners, Reviewers, other team members)
* Priority
* Repository (e.g. Git Repository URL)

### Activity

* Id
* Label (a human readable text describing the activity)
* Type
* StartTime: *optional*
* EndTime: *optional*
* Attributes: *optional*

### File

* Id
* Name
* Type
* Location (unique URL to locate the file in GIT)
* Owner
* Last modified
* Status (Out-of-date, QC failed,…)

### Assumption

* Type
* Body
* Justification
* Established
* Testable
* TestApproach
* TestOutcome
* Metadata

### Metadata

* Field
* Value

### Decision

* Description

# Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

* Describes a significant business need
* Identifies, documents, and ranks the problem that is driving the scenario
* Describes the business and technical environment that will resolve the problem
* States the desired objectives
* Shows the “Actors” and where they fit in the business model
* Is specific, and measurable, and uses clear metrics for success

All use cases define a goal, a responsible actor, preconditions, a basic flow, alternate flows, exceptions, and post conditions. A goal, a responsible actor and a basic flow are mandatory for a use case the other attributes are optional and should only be defined if needed.

Exceptions and alternate flows should be described like entry points to the basic flow. As a good practice the identifier of an exception or alternate flow should reflect the entry point to the basic flow (i.e.: AF3 – **A**lternate **f**low that could happen during step **3** of the basic flow; Ex4 – **Ex**ception that could happen during step **4** of the basic flow; AFx – **A**lternate **f**low that could happen at any time during the basic flow).

## Actors / User roles

These actors are used throughout this document to represent people and there functions that interact with the software system.

### Scientist

A Scientist can access projects, assigned files and perform activities on projects and the assigned files

### Manager

A Manager is responsible for a Team of Scientists and can access their work readonly

### Reviewer

Reviewer is not a user role. The classification as Reviewer depends on the project and is defined by the project owner.

## UC1 – Login

|  |  |
| --- | --- |
| Goal: | A user can sign into the web application for performing actions to projects. |
| Actor: | Scientist, Manager |
| Preconditions: | The User is not already signed in. |
| Basic Flow: | 1. Enter Username 2. Enter Password 3. Click button “Sign in” |
| Alternate Flow: | - |
| Exceptions: | Login failed   1. Username not existing 2. Wrong password to username 3. No username and/or password was given |
| Post Conditions: | On successfully signed in the user is redirected to his projects page. |

## UC2 – Create a project

|  |  |
| --- | --- |
| Goal: | Create a new project |
| Actor: | Scientist |
| Preconditions: | The user is signed in |
| Basic Flow: | 1. Select “New” on the projects page. 2. A modal dialog is shown to enter the project details. 3. Enter name for new project. 4. Enter the project priority. 5. Enter the GIT Repository URL 6. Click “Create” to confirm |
| Alternate Flow: | AF5: Create a new Git Project by using the DDMoRe infrastructure  AF6: Cancel creation of new project by closing the modal dialog. |
| Exceptions: | The GIT Repository URL of the project is not unique over all users.   1. An error message will be provided to the user in the modal dialog. The user can now change the new GIT Repository URL so the it is unique or close the dialog to cancel new project creation. |
| Post Conditions: | The new created project will be shown to edit data. |

## UC3– Show own projects

|  |  |
| --- | --- |
| Goal: | Show all projects where the signed in user is the owner |
| Actor: | Scientist |
| Preconditions: | The user is signed in |
| Basic Flow: | Go to the projects page |
| Alternate Flow: | The user doesn’t have projects assigned.   1. Show information “No Projects found” |
| Exceptions: | - |
| Post Conditions: | The application shows a table of all projects. This table has the following data:   1. Project Name 2. Priority 3. Project Owner 4. Reviewers 5. Start date and time 6. End date and time (optional) |

## UC4 – Edit a project

|  |  |
| --- | --- |
| Goal: | Edit project metadata. |
| Actor: | Scientist |
| Preconditions: | The user is signed in |
| Basic Flow: | 1. Use Case: “Open Project” 2. Now you can do the following actions:    1. Edit metadata    2. Edit the priority    3. Edit Assumptions    4. Edit summary |
| Alternate Flow: | Execute no actions is the same as “Open project” |
| Exceptions: | - |
| Post Conditions: | The new data is immediately shown as part of the current selected project. |

## UC5 – Show projects by user

|  |  |
| --- | --- |
| Goal: | The Manager can access projects as readonly for Scientists. |
| Actor: | Manager |
| Preconditions: | The user is signed in |
| Basic Flow: | 1. Go to the projects page 2. Select the user |
| Alternate Flow: | The user doesn’t have projects assigned.   1. Show information “No Projects found” |
| Exceptions: | - |
| Post Conditions: | The application shows a table of all projects. This table has the following data:   1. Project Name 2. Priority 3. Project Owner 4. Reviewers 5. Start date and time 6. End date and time (optional) |

## UC6 – Show projects to review

|  |  |
| --- | --- |
| Goal: | A user can query all projects which have him assigned as reviewer. |
| Actor: | User |
| Preconditions: | The user is signed in |
| Basic Flow: | 1. Go to the projects page 2. Select “Review Projects” |
| Alternate Flow: | The user doesn’t have projects assigned.   1. Show information “No Projects found” |
| Exceptions: | - |
| Post Conditions: | The application shows a table of all projects. This table has the following data:   1. Project Name 2. Project Owner 3. Reviewers 4. Start date and time 5. End date and time (optional) |

## UC7 – Open Project

|  |  |
| --- | --- |
| Goal: | A user can open a project to see all assigned data. |
| Actor: | User |
| Preconditions: | The user is logged in |
| Basic Flow: | 1. Go to the projects page 2. Select a project |
| Exceptions: | - |
| Post Conditions: | The application shows a project data on the page. |

## UC8 – Create Assumption

|  |  |
| --- | --- |
| Goal: | The scientist can create one to many assumptions for entities |
| Actor: | Scientist |
| Preconditions: | The user is signed in and is the owner of the selected project |
| Basic Flow: | 1. Create an Assumption    1. Type    2. Body    3. Justification    4. Established    5. Testable    6. Test Approach    7. Test Outcome 2. Save the assumption |
| Alternate Flow: | Cancel save   1. Changes are discarded |
| Exceptions: | - |
| Post Conditions: | A file that contains the assumptions is created and visualized in the project detail page. The new data is immediately shown as part of the current selected project. |

## UC9 – Modify Assumption

|  |  |
| --- | --- |
| Goal: | The scientist can modify an existing assumption |
| Actor: | Scientist |
| Preconditions: | The user is logged in and is the owner of the selected project |
| Basic Flow: | 1. Select an Assumption. Change on or many Attributes of the assumption    1. Type    2. Body    3. Justification    4. Established    5. Testable    6. Test Approach    7. Test Outcome 2. Save the assumption |
| Alternate Flow: | Cancel save   1. Changes are discarded |
| Exceptions: | - |
| Post Conditions: | A new version of the assumption is created and the relationships:   * “wasDerivedFrom” to the previous version is created. * “update assumption” that tracks who made when the modification   The new data is immediately shown as part of the current selected project. |

## UC10 – Invalidate Assumption

|  |  |
| --- | --- |
| Goal: | The scientist can set a hide flag to an existing assumption. |
| Actor: | Scientist |
| Preconditions: | The user is logged in and is the owner of the selected project |
| Basic Flow: | 1. Select an Assumption 2. Mark Assumption to invalidated 3. Ask for confirmation 4. Save changes |
| Alternate Flow: | Cancel deletion   1. Changes are discarded – Post Conditions are skipped |
| Exceptions: | - |
| Post Conditions: | The assumption is invalidated. The new data is immediately shown as part of the current selected project. |

## UC11– Create Decision

|  |  |
| --- | --- |
| Goal: | The scientist can create one to many decisions based on project entities. |
| Actor: | Scientist |
| Preconditions: | The user is logged in and is the owner of the selected project |
| Basic Flow: | 1. Select the entities upon which this decision should be based 2. Create a Decision and provide a description for the decision 3. Save the Decision |
| Alternate Flow: | Cancel save   1. Changes are discarded |
| Exceptions: | - |
| Post Conditions: | A file that contains the decision is created and visualized in the project detail page. The new data is immediately shown as part of the current selected project. |

## UC12– Modify Decision

|  |  |
| --- | --- |
| Goal: | The scientist can modify an existing decision |
| Actor: | Scientist |
| Preconditions: | The user is logged in and is the owner of the selected project |
| Basic Flow: | 1. Select the decision and modify 2. Save |
| Alternate Flow: | Cancel save   1. Changes are discarded |
| Exceptions: | - |
| Post Conditions: | A new version of the decision is created and a relationships:   * “wasDerivedFrom” to the previous version is created. * “update assumption” that tracks who made when the modification   The new data is immediately shown as part of the current selected project. |

## UC13 – Invalidate Decision

|  |  |
| --- | --- |
| Goal: | The scientist can set a hide flag to an existing decision. |
| Actor: | Scientist |
| Preconditions: | The user is logged in and is the owner of the selected project |
| Basic Flow: | 1. Select a Decision 2. Mark Decision to invalidated 3. Ask for confirmation 4. Save changes |
| Alternate Flow: | Cancel deletion   1. Changes are discarded – Post Conditions are skipped |
| Exceptions: | - |
| Post Conditions: | The Decision is hided. The new data is immediately shown as part of the current selected project. |

## UC14 – Show file content

|  |  |
| --- | --- |
| Goal: | The user can open a file and see the file content in a readonly view. |
| Actor: | Scientist, Manager, Reviewer |
| Preconditions: | 1. The user is logged in 2. The Scientist is owner of the project 3. The manager can open the scientist project 4. The Reviewer is reviewer for this project |
| Basic Flow: | 1. Open a project 2. Select a File that contains text content. 3. Choose “Open File” 4. Ask for confirmation if the file content exceeds 100 kb. The amount of kb should be configurable. 5. The application shows the content of the file in a readonly view |
| Alternate Flow: | Cancel open   1. The File is not opened – post conditions are skipped |
| Exceptions: | - |
| Post Conditions: | The file content is displayed in a readonly text view. |

## UC15 – Tag a Model

|  |  |
| --- | --- |
| Goal: | Tag a Model as:   * Base Model * Final Model * Pivotal Model |
| Actor: | Scientist |
| Preconditions: | The user is logged in |
| Basic Flow: | 1. Select a Model 2. Apply tag |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | The tag is applied to the selected model. |

## UC16 – Remove tag from a model

|  |  |
| --- | --- |
| Goal: | Remove a tag from a model |
| Actor: | Scientist |
| Preconditions: | The user is logged in |
| Basic Flow: | 1. Select a model 2. Remove tag 3. Ask for confirmation |
| Alternate Flow: | Cancel deletion   1. Changes are discarded – Post Conditions are skipped |
| Exceptions: | - |
| Post Conditions: | The tag is removed from the model. |

## UC17 – Set QC Flag

|  |  |
| --- | --- |
| Goal: | Set QC status (true = passed QC/ false = failed QC). |
| Actor: | Reviewer |
| Preconditions: | The user is logged in and is a reviewer of the selected project |
| Basic Flow: | 1. Select one to many entities 2. Set QC status passed or failed |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | In case of false all related entities are invalid as well. |

## UC18 – Export Project

|  |  |
| --- | --- |
| Goal: | Export project data as a ZIP file. |
| Actor: | Scientist, Manager |
| Preconditions: | The user is logged in |
| Basic Flow: | 1. Select a project 2. Select “Export” |
| Alternate Flow: | - |
| Exceptions: |  |
| Post Conditions: | The export is a ZIP file containing the following files:   1. A XML file containing all project data description (e.g. input files, output files, assumptions, …) 2. All project related files. |

## UC19 – Copy Table

|  |  |
| --- | --- |
| Goal: | For every HTML table copy the content into clipboard for pasting it into e.g. Word or Excel. |
| Actor: | Scientist, Manager |
| Preconditions: | The user is logged in |
| Basic Flow: | Select a table and click export. |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | The table content is available from the clipboard. |

## UC20 – Audit Trail

|  |  |
| --- | --- |
| Goal: | Show chart visualization for an entity. |
| Actor: | Scientist, Manager |
| Preconditions: | The user is logged in |
| Basic Flow: | 1. Open a project 2. Select an entity 3. Click on “Audit Trail” |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | A new browser tab (window) will open with the chart. |

## UC21 – Run record

|  |  |
| --- | --- |
| Goal: | Create table of key model development steps, suitable for inclusion in a report. The table will contain filename of the model, description, and fields parent, OFV, deltaOFV, condition number, comment (populated from run file, provenance database, or if not available, blank.) |
| Actor: | Scientist, Manager |
| Preconditions: | The user is logged in; key runs have been tagged in the current project |
| Basic Flow: | 1. Open a project 2. Click on “Run record” |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | A new browser tab (window) will open with the table. |

## UC22– QC checklist

|  |  |
| --- | --- |
| Goal: | Create QC checklist table, suitable for use by a reviewer. The table will contain all key steps (as tagged by the Scientist), as well as all forward and backward dependencies of these as determined by traversing the provenance database. |
| Actor: | Scientist, Reviewer, Manager |
| Preconditions: | The user is logged in; key runs have been tagged in the current project |
| Basic Flow: | 1. Open a project 2. Click on “QC checklist” |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | A new browser tab (window) will open with the table. |

## UC23 – Assumptions table

|  |  |
| --- | --- |
| Goal: | Create assumptions table, suitable for inclusion in a report. The table will contain all fields recorded within each assumption object related to “key steps”, as well as the key step to which each assumption is related. See Marshall et al Good Practices in Model‐Informed Drug Discovery and Development: Practice, Application, and Documentation. CPT:PSP 2016 Mar 1;5(3):93–122. |
| Actor: | Scientist, Reviewer, Manager |
| Preconditions: | The user is logged in; key runs have been tagged in the current project; assumptions have been defined in the current project |
| Basic Flow: | 1. Open a project 2. Click on “Assumptions table” |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | A new browser tab (window) will open with the table. |

## UC24 – Assign or change of reviewer(s)?

|  |  |
| --- | --- |
| Goal: | A user can assign or change reviewer(s) to a project. |
| Actor: | Scientist, Manager |
| Preconditions: | The user is logged in; a project has been created; there are steps within the current project. |
| Basic Flow: | 1. Go to the projects page 2. Select the project for review 3. Unselect a user if there is one already assigned 4. Select the user(s) to assign for review (or type in the name of user?) 5. Save |
| Alternate Flow: | Cancel save   1. Changes are discarded. |
| Exceptions: | - |
| Post Conditions: | The updated list of reviewers displayed in the projects page. |

## UC25 – Assign or Update a reviewer

|  |  |
| --- | --- |
| Goal: | A user can access projects as read-only for Scientists. |
| Actor: | Manager |
| Preconditions: | The user is signed in |
| Basic Flow: | 1. Go to the projects page 2. Select the user |
| Alternate Flow: | The user doesn’t have projects assigned.   1. Show information “No Projects found” |
| Exceptions: | - |
| Post Conditions: | The application shows a table of all projects. This table has the following data:   1. Project Name 2. Project Owner 3. Reviewer 4. Start date and time 5. End date and time (optional) |

## UC26– Decision table

|  |  |
| --- | --- |
| Goal: | Create decision table, extract the text in Decision for all Steps in the current Project. |
| Actor: | Scientist, Reviewer, Manager |
| Preconditions: | The user is logged in; key steps have been tagged in the current project; decision have been defined in the current project |
| Basic Flow: | 1. Go the projects page 2. Select the project 3. Select the Steps 4. Click on “Decision table” |
| Alternate Flow: | - |
| Exceptions: | - |
| Post Conditions: | A new browser tab (window) will open with the table. |