Ecosystem infrastructure for smart and personalised inclusion and PROSPERITY for ALL stakeholders

D203.4 Clinician/Consumer Custom Solution Development environment

Project Acronym	**Prosperity4All**
Grant Agreement number	**FP7-610510**
Deliverable number	**D203.4**
Work package number	**WP203**
Work package title	**Collaborative development tools / environments**
Authors	**Stefan Parker(KI-I), Gerhard Nussbaum (KI-I)**
Status	**Final**
Dissemination Level	**Public**
Delivery Date	**06.07.2017**
Number of Pages	**18**

Keyword List

WebACS, ACS, AsTeRICS, Wizard

Version History

Revision	Date	Author	Organisation	Description
1	30/05/2017	Stefan Parker	KI-I	first draft
2	07/06/2017	Stefan Parker	KI-I	ready for peer review
3	28/06/2017	Stefan Parker	KI-I	peer comments incorporated, ready for coordinators review
4	06/07/2017	Stefan Parker	KI-I	coordinators comments incorporated, ready for submission
5	10/07/2017	Stefan Parker	KI-I	Alternative texts for images updated, deliverable finalised

Table of Contents

List of Figures

List of Abbreviations

Abbreviation	Full form
ACS	AsTeRICS Configuration Suite
ARE	AsTeRICS Runtime Environment
AsTeRICS	Assistive Technology Rapid Integration & Construction Set
AT	Assistive Technology
CCCSD	Clinician/Consumer Custom Solution Development environment (CCCSD)
DSpace	DeveloperSpace
IT	Information Technology
KI-I	Kompetenznetzwerk Informationstechnologie zur Förderung der Integration von Menschen mit Behinderungen
P4A	Prosperity4all
REST	Representational State Transfer
UCY	University of Cyprus
WebACS	Web-based AsTeRICS Configuration Suite

Executive Summary

This deliverable is about the prototype of the clinician/consumer custom solution development environment. It consists of a database of AsTeRICS models and a step-by-step wizard for choosing a suitable model. It enables people without a technical background to profit from the AsTeRICS framework.

1 Contribution to the global architecture

The technology reported here supports the usage of AsTeRICS by people with no technical background. It is therefore, the same as AsTeRICS, part of the DeveloperSpace “building blocks” and “frameworks".

The clinician/consumer custom solution development environment (CCCSD) is a wizard based tool that enables people with no technical background (e.g. therapists, carers, family members, etc.) to find and deploy AsTeRICS models that are suitable for their clients either directly, or with minor adaptations. AsTeRICS provides a mechanism for visually combining functional units on screen that allows a person without programming skills to be able to create new assistive technology configurations. The CCCSD goes one step further and opens this world up for people with no technical background at all.

This also will help to push development work out to the consumer and service-delivery professionals, who can help to fill the associated database with more models and therefore make the wizard an increasingly useful tool for their technically less skilled employees.

It also opens up possibilities for engaging younger boys and girls in programming and assistive technology by providing tools that are more reachable and useable by them at an early age.

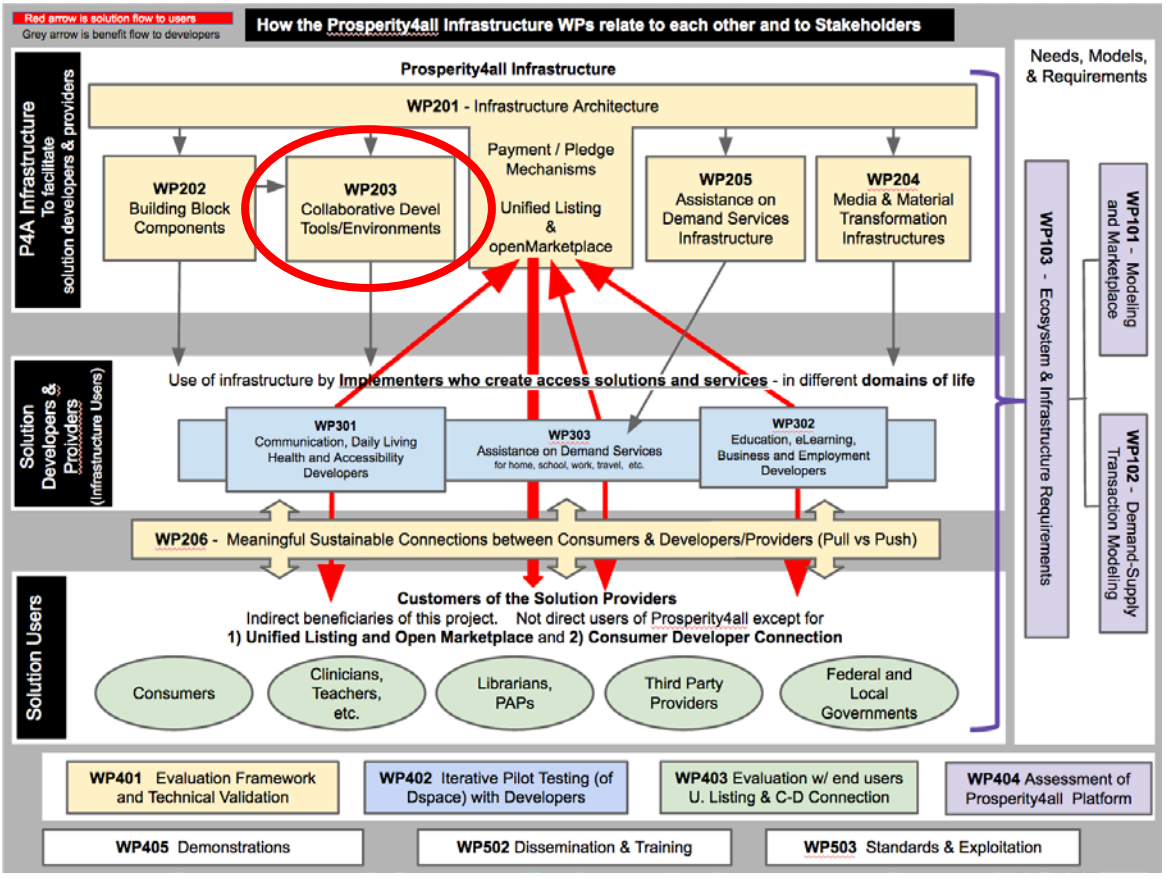

Figure 1: Overall Picture of Prosperity4all

2 Concept

2.1 Initial Situation

The AsTeRICS Configuration Suite (ACS) – the "old" version [1] developed during the AsTeRICS project [2] (Grant Agreement No.247730), as well as the newer version developed in the Prosperity4All project, based on web technologies (WebACS [3]) – is a tool for constructing customised Assistive Technology (AT) solutions by connecting components through data- and event-channels in a graphical editor and by adjusting their properties. The usage of the ACS requires no programming skills, thus enabling non-programmers to build individually tailored AT. However, it does require a certain amount of technical skill. Therapists and carers as well as family members or even end users themselves often do not have these technical skills and therefore cannot use the ACS (or at least not to its full extent).

2.2 Goals and Design Targets

The CCCSD enables people with little or no technical skill to profit from the strengths of AsTeRICS without having to extend their technical knowledge. Also it helps to foster the work of ACS users by giving them a shortcut towards pre-existing models and therefore shortening model development time significantly. For this purpose a step-by-step wizard has been created that helps to choose a predefined model from a database. The decision for a certain model is based on basic information:

- Device category: what does the user want to control?
- Technical prerequisites: what hardware is available or can/will be bought?
- Body functions: what body functions can be used or are preferred?

The chosen model can then be either directly deployed to the AsTeRICS Runtime Environment (ARE) or opened in the ACS to set some properties (for those users capable of doing so). Additionally it is possible to deploy the chosen model directly to the ARE, so that it can be used right away.

3 Architecture

The CCCSD consists of two major parts: a web-hosted model repository (backend) and the actual wizard (frontend).

3.1 Model repository

The CCCSD model repository is a constantly growing database of AsTeRICS models, for which the original developer has defined certain metadata that help identify the usefulness of this specific model for a certain user. These data are:

- **Model name**
- **Model description**: What does the model do?
- **Device category**: What will the end user want to control, e.g. computer, environment, etc.?
- **Body functions**: Which function(s) of the users body can or will be used for input, e.g. head, arm, muscle signals, etc.?
- **Technical prerequisites**: What hardware is already available or what is the user ready to buy, e.g. webcam, button, etc.?

The repository also has its own **user administration**. Only registered users can upload models to the database. A user has to give his name and email address to be able to register and can then define a username and a password. Each user has a certain **role** – "normal user" or "admin" - default is "normal user". An admin has the right to assign admin rights to another user. Also the admin is allowed to set the **"approved"**-flag for models uploaded by a normal user. This means that this model has been checked by an expert and approved as useful.

The database has been implemented in MySQL [4]. Figure 2 shows the schema of this database.

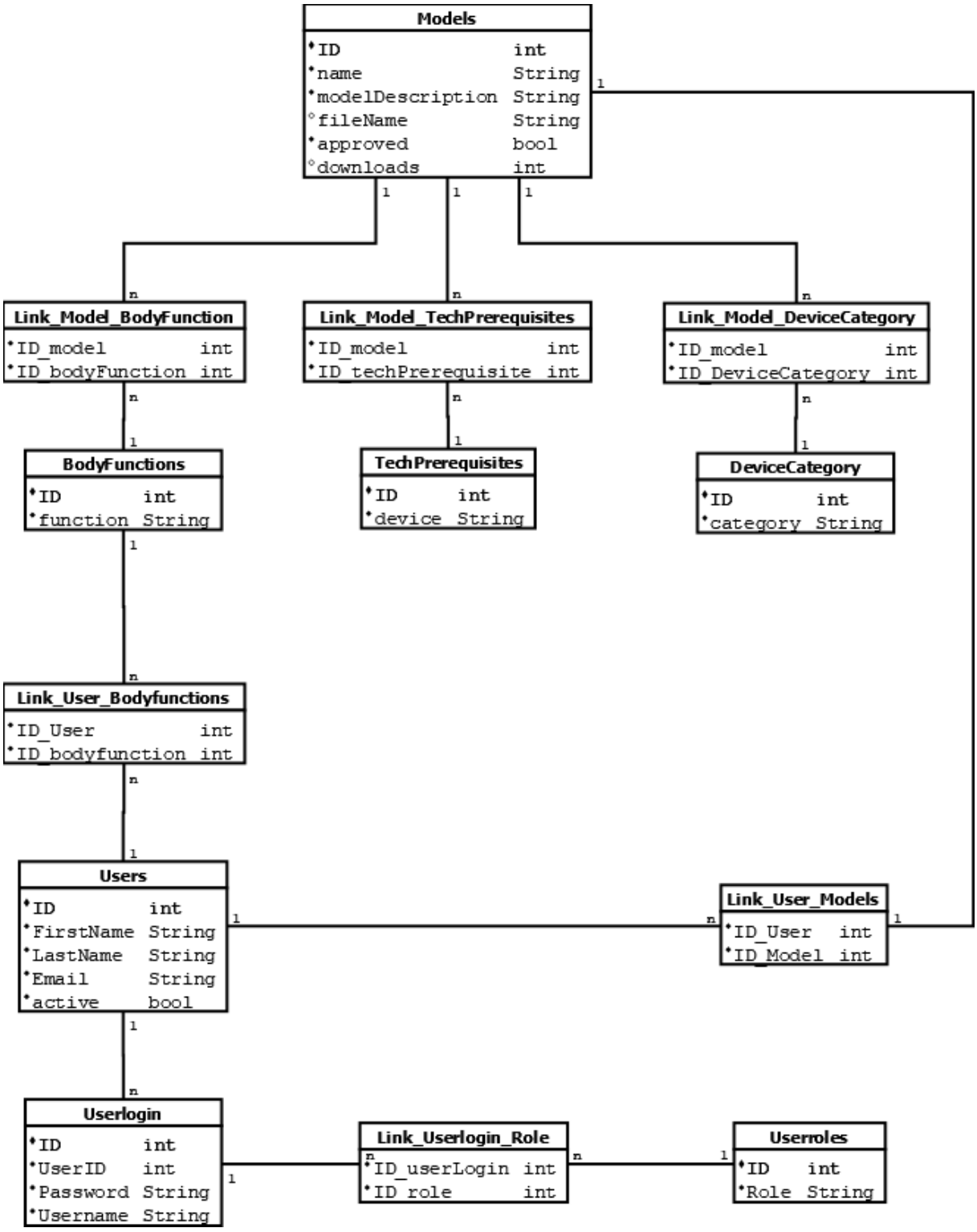

Figure 2: Database schema of model repository

3.2 Wizard

The step-by-step wizard has been implemented in Javascript and comprises the following steps:

1. **What** do you want **to control**?
2. What **hardware** do you have available? (this step can be skipped, if the user does not care what hardware shall be required for the model)
3. What **body functions** can you / do you want to use?

The wizard then gives the user a list of suitable models, the details of which can be viewed by clicking the model name. For each model three options will be provided:

- Download: downloads the model file in .acs-format to the local hard drive
- Send to ACS: opens the model in the WebACS
- Send to ARE: directly deploys the model to the ARE, ready to use

For the communication between the wizard and the ARE the REST interface provided by partner UCY is used.

4 Implementation

The prototype of the CCCSD is available on GitHub: **https://github.com/asterics/CCCSD**

4.1 Model repository

The backend has been implemented server-side using PHP (version 5.6.23). Upon login the first view is the “My Models” panel, showing all models this user has already uploaded (and consequently can update or delete) (see Figure 3).

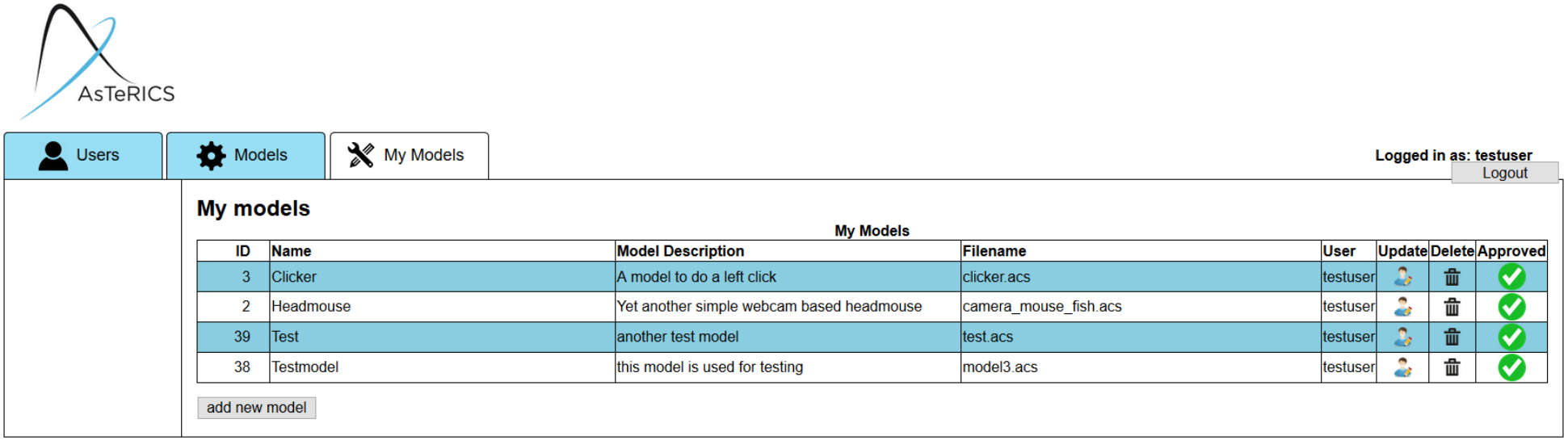

Figure 3: My Models

Via the “add new model” button, a new model can be uploaded, described, and the parameters “Tech-Prerequisites”, “Device Categories” and “Bodyfunctions” can be set.

The first view in the model tab is rather similar, only that it shows all available models, no matter who has uploaded them. The model list in Figure 4 contains one model that has been uploaded by a normal user and which has not yet been approved by an admin. The columns for “Update” and “Delete” are missing in this picture, because logged-in user (user “mmust”) is a normal user, who does not have the right to alter other people’s models. An admin, however, does have this right.

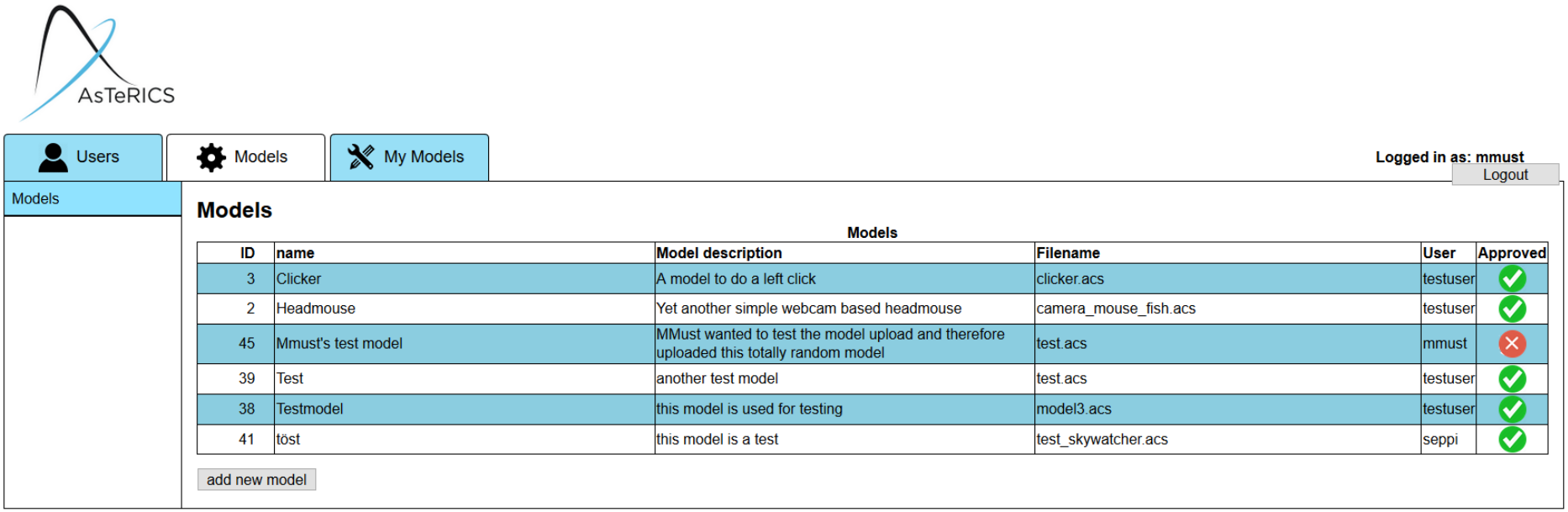

Figure 4: Models Tab

When logged in as admin, a user also has three more entries in the submenu at the left: “Tech-Prerequisites”, “Bodyfunctions” and “Device Categories”. These are only available for admin users and allow adding new entries for the respective tables. Normal users can only use the existing entries or contact an admin and ask additional ones to be added. Figure 5 shows an admin (user “testuser”) on the Tech-Prerequisites table. Note that the “Delete” option is only available for two of the entries. That is because the other entries have already been used in one or several models and consequently may not be deleted anymore in order to avoid inconsistencies. This works in the same way for “Bodyfunctions” and “Device Categories”.

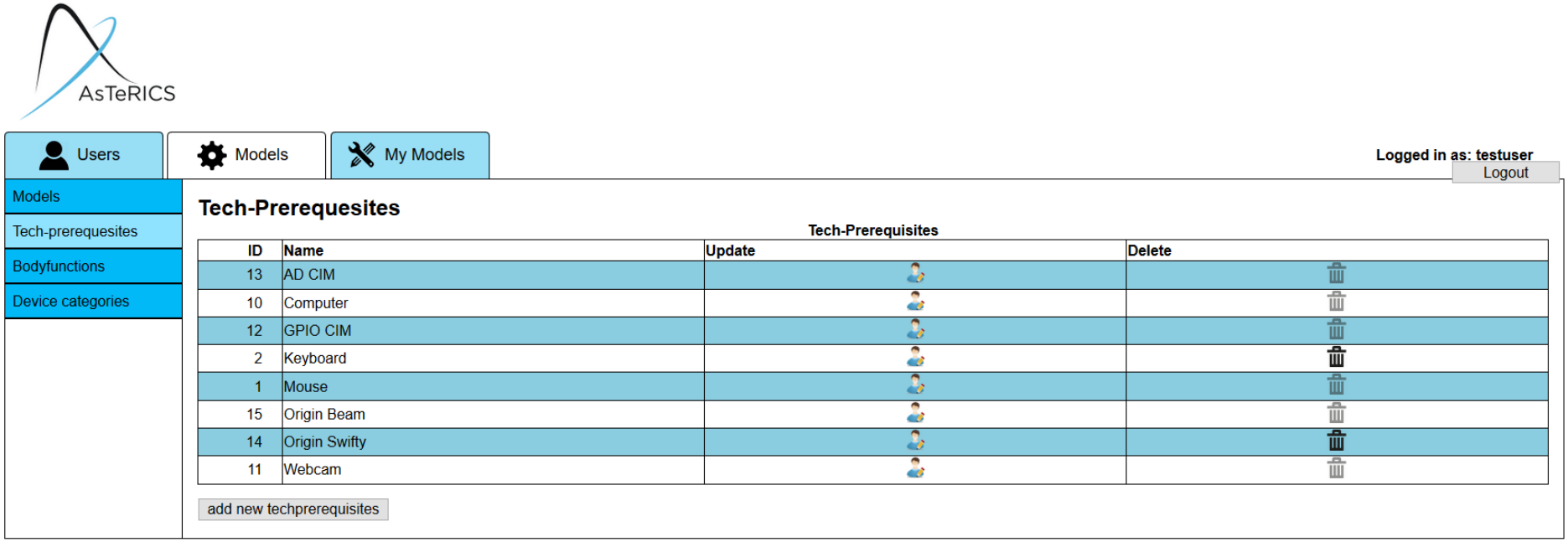

Figure 5: Tech-Prerequisites

The user tab is rather unspectacular for normal users, since all they will see is their own account, with the options to either update their information or delete their own account, which of course will result in immediate log out. Admin users will at this point see all registered users and have the right to alter or delete all accounts (Figure 6).

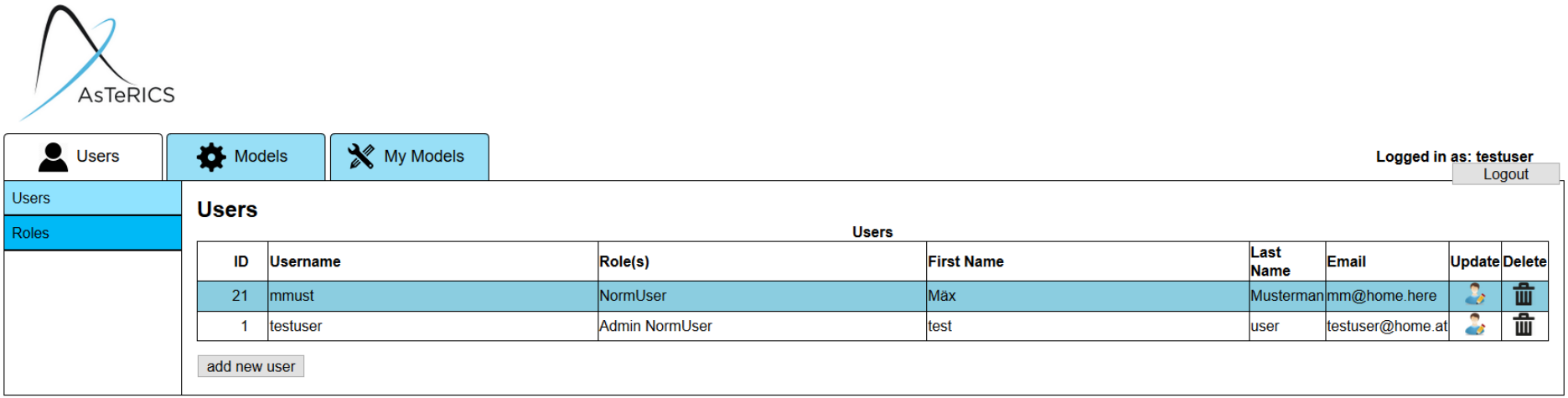

Figure 6: Users

4.2 Wizard

The wizard is implemented in Javascript, while using PHP for the database connection. The user is first welcomed by a start-page, offering a button to actually start the wizard (Figure 7). Via the “admin”-link in the footer the user can enter the backend.

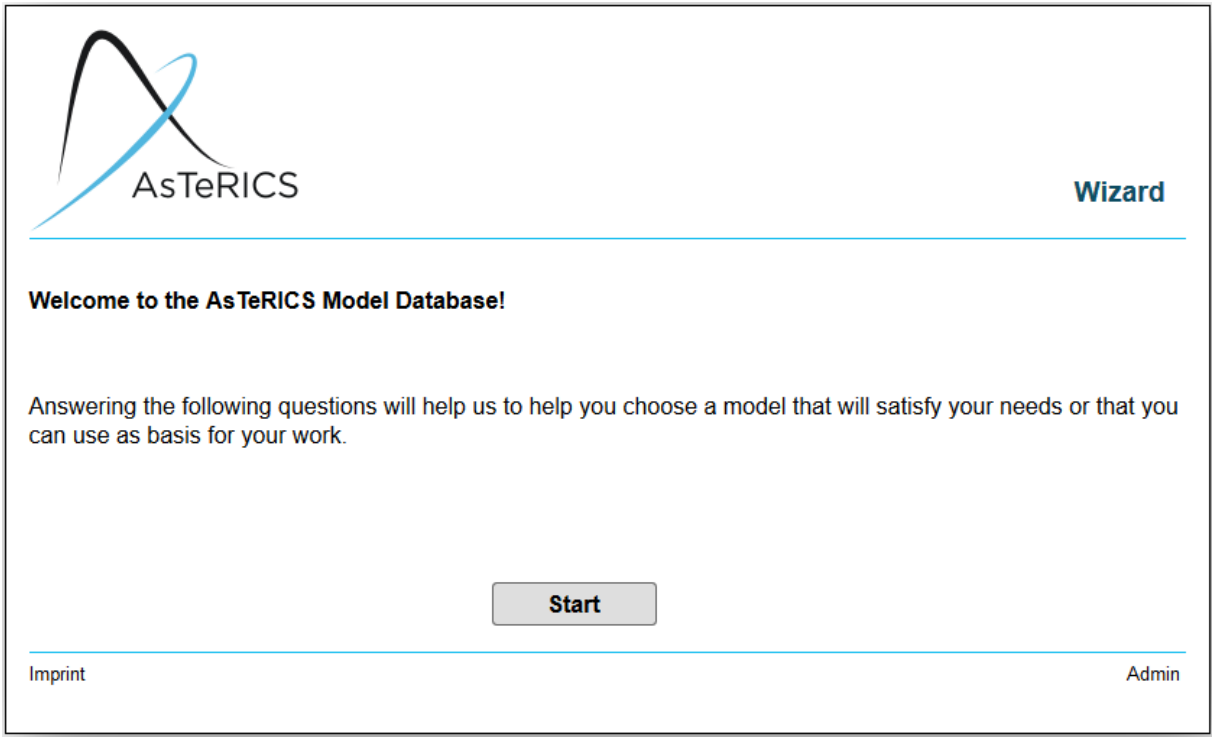

Figure 7: Wizard start-page

The wizard then asks the user to choose the devices categories he or she wants to control, followed by the technical prerequisites and the body functions. The user can select options by simply ticking the checkboxes provided by the wizard. It is also possible to switch back and forward between the questions to check or alter the choices already made. Figure 8 shows the question on “Body Functions” as an example.

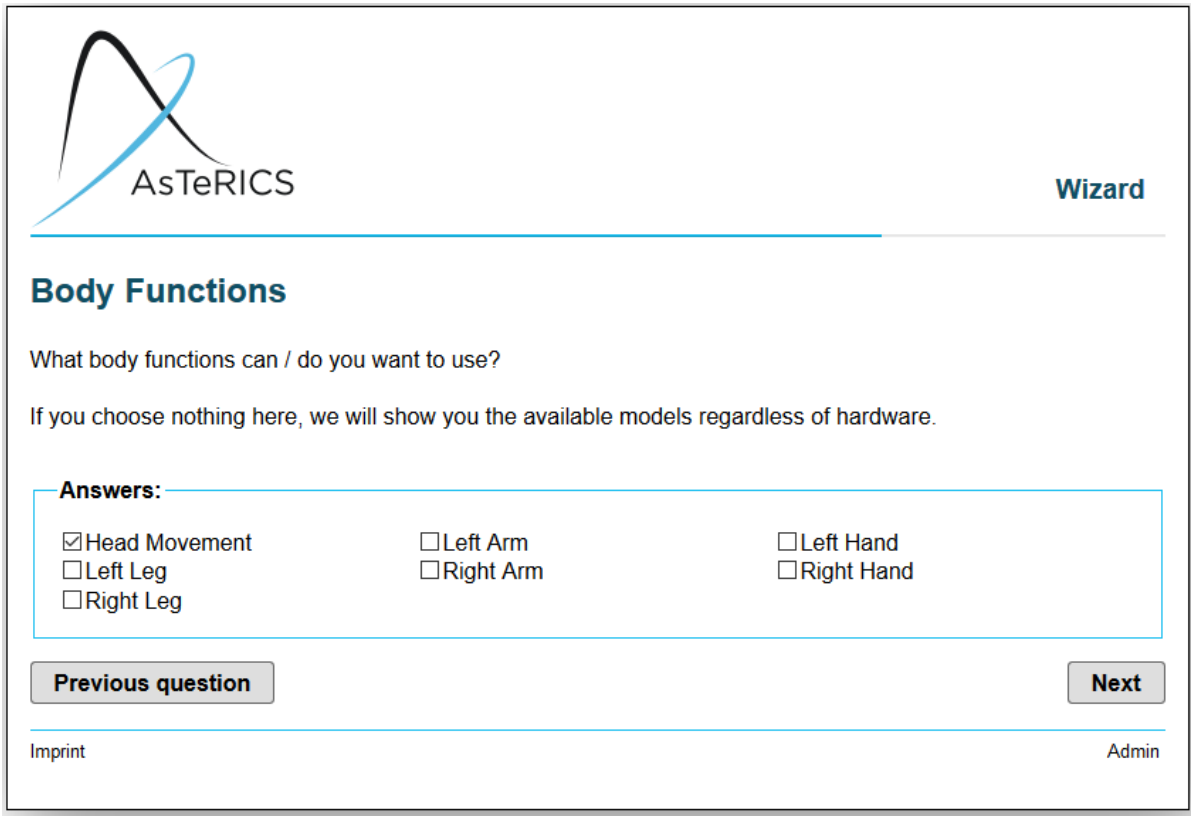

Figure 8: Wizard on "Body Functions"-page

After the user has answered all the questions, the wizard will show a list of available models. In the example in Figure 9, there are three models available for the entered parameters.

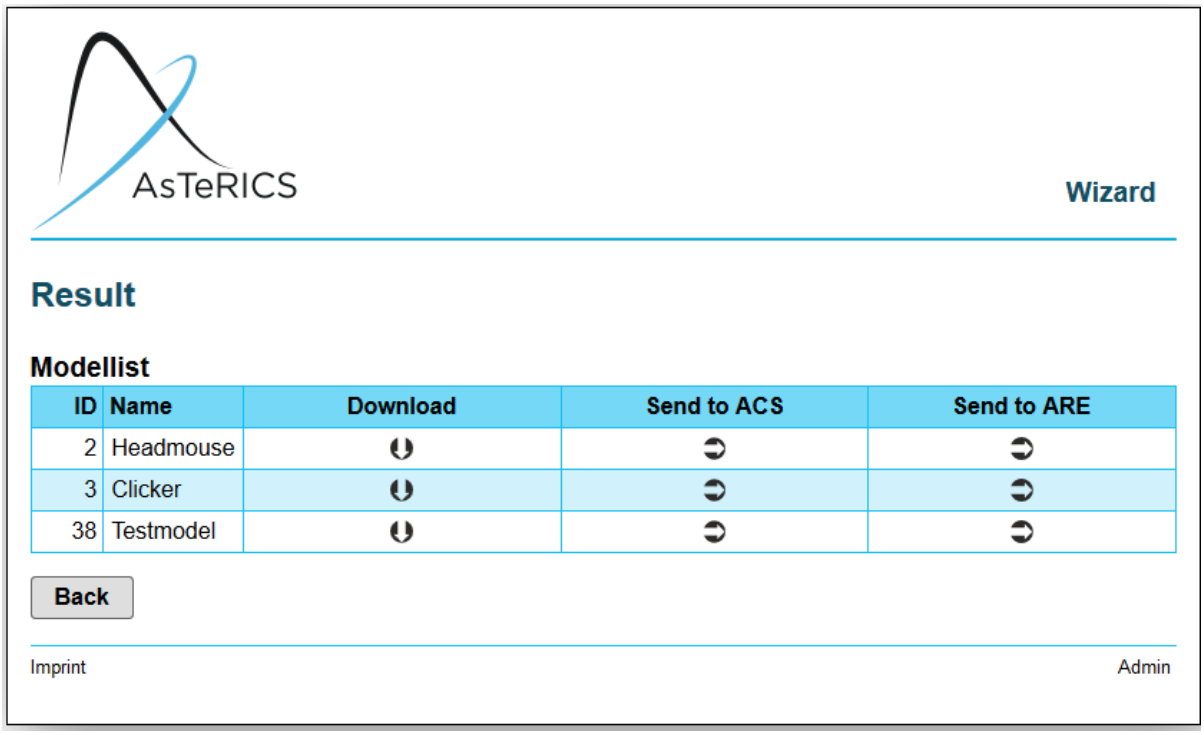

Figure 9: List of suitable models

From here the user can either directly use one of the provided options (download, send to ACS or send to ARE) or click on the model's name to show more details (Figure 10). The

details page gives more information on the model, like the description and who has uploaded it. It also provides the same options for model usage as were available in the list view.

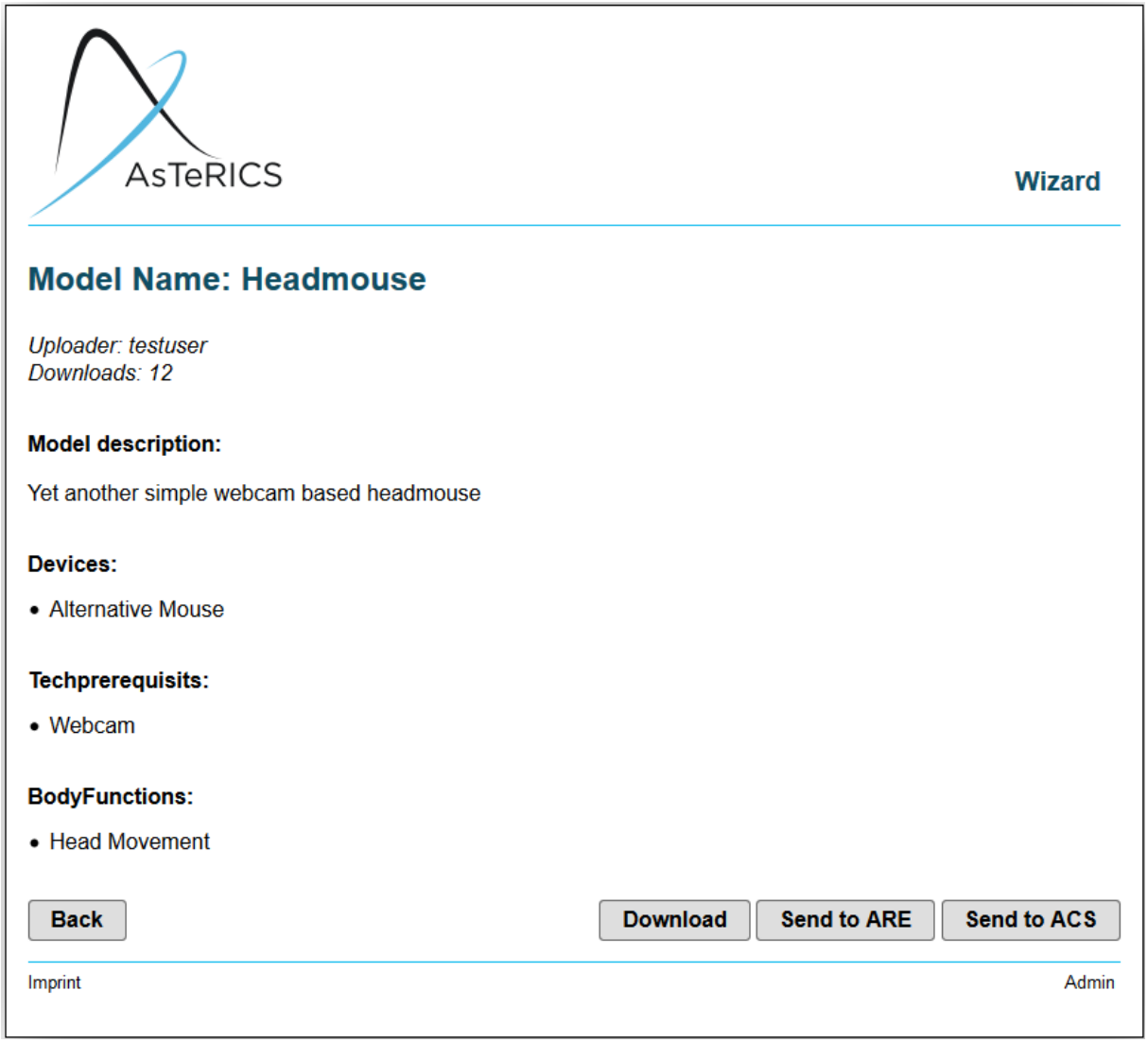

Figure 10: Detailed model view

5 References

[1] https://github.com/asterics/AsTeRICS

[2] http://www.asterics.eu

[3] https://github.com/asterics/WebACS

[4] https://www.mysql.com