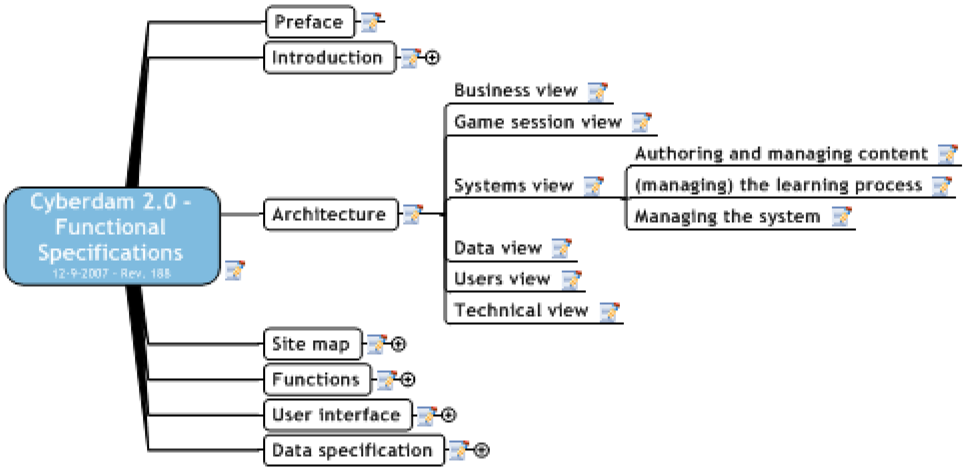
Cyberdam 2.0 - Functional Specifications



Final - 12 September 2007 - Pieter van der Hijden (pvdh@sofos.nl)

Main changes with respect to Cyberdam 1.1:

- migration of the platform from OpenACS to Java,
- simplification of the gameplay mechanism (earlier called workflow) by introducing a new activity type, the progress,
- simplification of the enrolment mechanism (earlier called case instantiation),
- more consistent user interface.

Main changes with respect to earlier draft specification:

- game authoring will be part of the system, the idea to use a scorm editor for authoring is left for the moment
- combination of game (roles) and playground (objects) will be called "game manifest".

Main changes with respect to draft 22nd of July:

- architecture
    - for each data object it has been made clear who the owner is; the role of user administrator was added
- site map
    - the names of the pages have been harmonised
    - the description of all pages has been completed
    - the sequence of the pages is from game development to game delivery (earlier: the other way round)
    - the concept of parallel sessions (of the same game) as it exists in Cyberdam 1.1, was no longer needed, as sessions can be copied easily (and updated where necessary).

To be discussed:

- should there be a separate user role for dialogue text management?

Still missing:

- What should be logged at game session level, at system level?
- The mechanism to set and send e-mail notifications:
    - default value for notifications during sessions (e-mail received, new step- of-play reached)
    - actual value for notifications during a certain session (e-mail received, new step- of-play reached)
- The mechanism to handle stylesheets.
- The mechanism to let the playground objects drive the city map objects.
- The information in the Site Map chapter could partially be moved to the Functions chapter and to the Data chapter.

Main changes with respect to draft 31 July 2007:

- conclusions of e-mail correspondence by Simon Groenewolt and Pieter van der Hijden during August 2007, a.o. the "still missing" items
- site map chapter, an overview picture has been added, the sequence of the paragraphs has been changed
- small lay-out improvements and typo corrections

Table of Contents

1 Preface

Cyberdam is an electronic learning suite for computer- supported asynchroneous gaming/simulations in the context of a virtual city. It enables teaching staff to maintain a virtual city (interactive map plus underlying websites) that forms the context for online Role Playing Games. Teaching staff themselves can develop these games. Cyberdam supports the enrolment of the participants into game sessions and runs the sessions themselves. The whole is Internet based. Users only need an Internet browser and Internet connection to be able to participate.

This report is the top-level single source of reference for the Cyberdam 2.0 functional specifications. Where this report might be incomplete or more details are required, the following earlier reports will apply:

- Sieberdam/ROCS Specifications for Design 1.0; Pieter van der Hijden, 30 December 2004, VSNU RechtenOnline,
- Sieberdam/ROCS User Manual; Pieter van der Hijden, 30 December 2004, VSNU RechtenOnline.

The discussions leading to these specifications have been documented in:

- Cyberdam, now and in the future, development directions; Pieter van der Hijden, 26 januari 2007.
- Aanzet tot lijst met mogelijke verbeterpunten Sieberdam/ROCS; Geertje Bekebrede, 29 januari 2007.
- LieVW voor student. docent, ontwikkelaar en beheerder; Pieter van der Hijden, 6 mei 2007.
- Verslag bespreking LieVW - IJsfontein d.d. 1 juni 2007; Vanessa Noordervliet, Robert Fischer, Pieter van der Hijden.

2 Introduction

This chapter recapitulates Cyberdam's history, describes the current project and introduces a few changes in terminology. It paves the way to the specification of Cyberdam 2.0.

2.1 Sieberdam/ROCS

In 2003, The Scottish Ardcalloch project by Paul Maharg inspired the Dutch VSNU Program RechtenOnline to fund three projects aimed at the development of a Dutch equivalent: 1. the Dutch virtual City of Sieberdam, 2. the e-learning suite for asynchroneous workflow based group simulations ROCS, and 3. the integrated Sieberdam/ROCS. Rudi Holzhauer (project management, content authorship, and design), Martin de Wit (Flash programming), Willem Mieras (visual design), and Sander Gellaerts (student-assistant) developed Sieberdam. Aernout Schmidt and Jeroen Leijen (project management and design), Lokman Tsui (design), Peter van Schijndel (design), and Collaboraid ApS København (software development) developed ROCS. Pieter van der Hijden (consultant to both projects) integrated Sieberdam and ROCS into Sieberdam/ROCS and extended the system (project management and design), together with Jarkko Laine (software development) and the former Sieberdam and ROCS project teams. Since January 2005, the Dutch Foundation RechtenOnline (Law-Online) is the owner of the product.

During 2005-2007 three projects applied Sieberdam/ROCS for education:

- SURFfoundation sponsored the KODOS project,
- Senter/Novem sponsored the ISOSIM project,
- Digitale Universiteit sponsored the project Virtuele Wijk (virtual quarter).

During these years, the software was not changed very much. Only in 2006, the project Digitale Wijk sponsored some improvements, resulting in the most recent software version: Sieberdam/ROCS 1.1.

2.2 LieVW project

From January 2007 till June 2009, the Dutch government program "Maatschappelijke Sectoren & ICT" will sponsor the so-called upscaling of Sieberdam/ROCS. This project, "Leren in een Virtuele Wereld (LieVW, in English: Learning in a Virtal World) aims at:

- developing, using and evaluating 25 new games based on Sieberdam/ROCS,
- upgrading the underlying software,
- establishing an organisation for continuing support and development, once the project has finished.

During the start-up phase of the LieVW project, it became clear that the current Sieberdam/ROCS software was insufficient stable, the OpenACS platform used less adequate, and the necessary expertise in TCL programming language hardly available. Continuing the use of this platform would imply continuing the existence of manifest or latent software problems, even after the completion of the LieVW project.

Migrating to another platform while improving the current software, would require a substantial effort, but was expected to be the most efficient solution in the long run. Project management decided therefore to follow this option. In cooperation with the project partners it selected the Java programming language and related open source software packages as the basis for the new platform. To mark this change and to make it easier to pronounce the name of the system in an international context, the name Sieberdam/ROCS was changed to Cyberdam.

2.3 Cyberdam

The development of Cyberdam will take place in subsequent rounds. Version numbering will start with 2.0.

Cyberdam 2.0 is, in short, Sieberdam/ROCS 1.1 based on the Java platform. More precisely, the Cyberdam 2.0 development effort implies:

- migration of the platform from OpenACS to Java,
- simplification of the gameplay mechanism (earlier called workflow),
- simplification of the enrollment mechanism (earlier called case instantiation),
- more consistent user interface.

Cyberdam 2.1, 2.2, 2.3 and possibly 2.4 will add new functionality to 2.0, partially based on existing requirements (like compatibility with international standards), partially based on improvements based on the evaluation of its application.

2.4 Terminology

For those who have been involved in the Sieberdam/ROCS projects, we summarise some new terms that will be used in Cyberdam 2.0 and its documentation. We tried to transform the earlier terminology from its object oriented programming and workflow management context to a gaming/simulation and electronic learning context.

The transformation table lists old and new terms.

old term	**new term**	**comment**
actor	participant	conforms to gaming/simulation context
case	game session	conforms to gaming/simulation context
case author	game (session) master	better fits the real function
portfolio	file directory	better fits the real function
simulation	either game model (the product) or game session (the process)	simply using the term "simulation" might be confusing
task	activity	better fits the real function
template	game (model)	conforms to gaming/simulation context
template author	game (model) author	conforms to e- learning context
yellow pages	city directory	avoids the use of a trademark

2.5 Overview of this report

This report describes:

- Architecture
- Site map
- Functions
- User interface
- Data specification

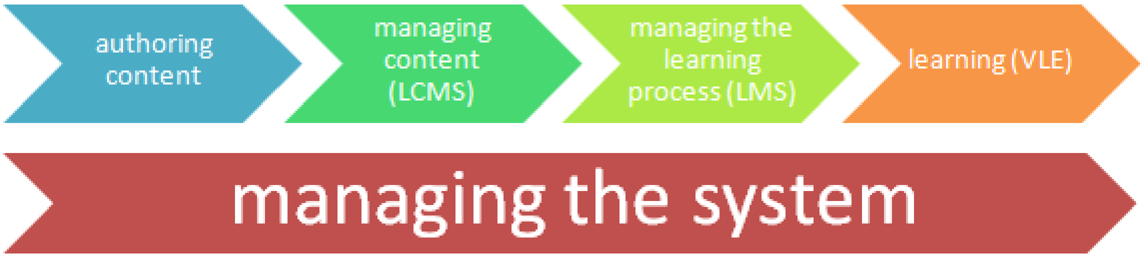
3 Architecture

The following sections describe the Cyberdam 2.0 architecture from five different perspectives:

- Business view
- Game session view
- Systems view
- Data view
- Users view
- Technical view

3.1 Business view

From a business point of view, a learning production process consists of four core processes: authoring content, managing content, managing/administering the learning process (course) and ultimately teaching/learning itself. The process of learning through games is not very different: authoring games as products, managing games as products, managing game sessions as processes, learning through games as processes. Cyberdam 2.0 therefore consists of four sub-systems to support these four core processes. Further, Cyberdam 2.0 has some additional functions at its disposal to manage the Cyberdam software system as a whole.



The four core processes are:

- Authoring content - Cyberdam supports creating and/or editing the game content: playground (objects), game models and the connection between these two: game manifests. Users involved in authoring the content are called playground (object) editors, game (model) authors, game manifest composers.
- Managing content (LCMS) - Cyberdam supports the storing and retrieving, importing and exporting of game content. In fact it acts as a Learning Content Management System (LCMS) for games. Users involved in managing content are called LCMS administrators or librarians.
- Managing the learning process (LMS) - Cyberdam supports the starting, administering and stopping of game sessions based on a game manifest and the handling of session logs and data produces during these sessions. In fact it acts as a Learning Management System (LMS) for game sessions. Users involved in managing learning processes are called LMS administrators. Users actually delivering learning processes through games are called game (session) masters.
- Learning (VLE) - Cyberdam supports the participating in a game session. In fact it acts as a Virtual Learning Environment (VLE) for game sessions. Users involved in managing the VLE are called VLE administrators. Users participating in games are called participants. Students, teaching staff and/or external experts can act as participants.

To make this all happen, the Cyberdam system itself has to be managed:

- Managing the system - Cyberdam supports a number of generic and/or housekeeping functions like user registration, multi-language support (mechanism to manage multiple dialogue language packs and to switch to another language) and e-mail notifications.

3.2 Game session view

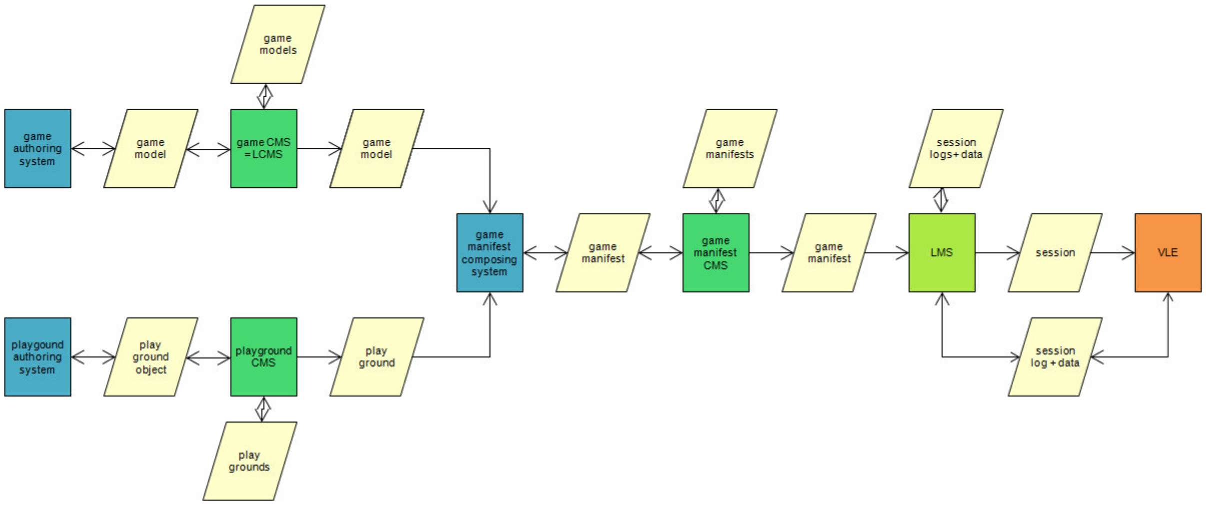
When a game master starts a game session, he starts a process that will follow a sequence of steps-of-play. This process is the same for all users participating to this game session, whether they are online or not.

For each step-of-play the game model prescribes which activities are enabled for which roles. When a user who is a participant of the game session logs in (i.e. starts a login-session), he sees the session home page for his role at the current step. It lists the activities that are enabled for this role at this step. Some activities will instruct the participant (i.e. the role) to send a message to another role. Other activities will instruct the role to upload a file to his online directory. A special activity is the progress activity. This activity moves the game session to the next (or another) step-of-play. Usually progress activities are allocated to roles played by staff.

The game session ends when it reaches a step-of-play that does not refer to any activities. This is called a normal end. A game master also can force a game session to end. This is called an abnormal end of the game session.

3.3 Systems view

The figure below describes the Cyberdam core processes as sub-systems connected by data flows.

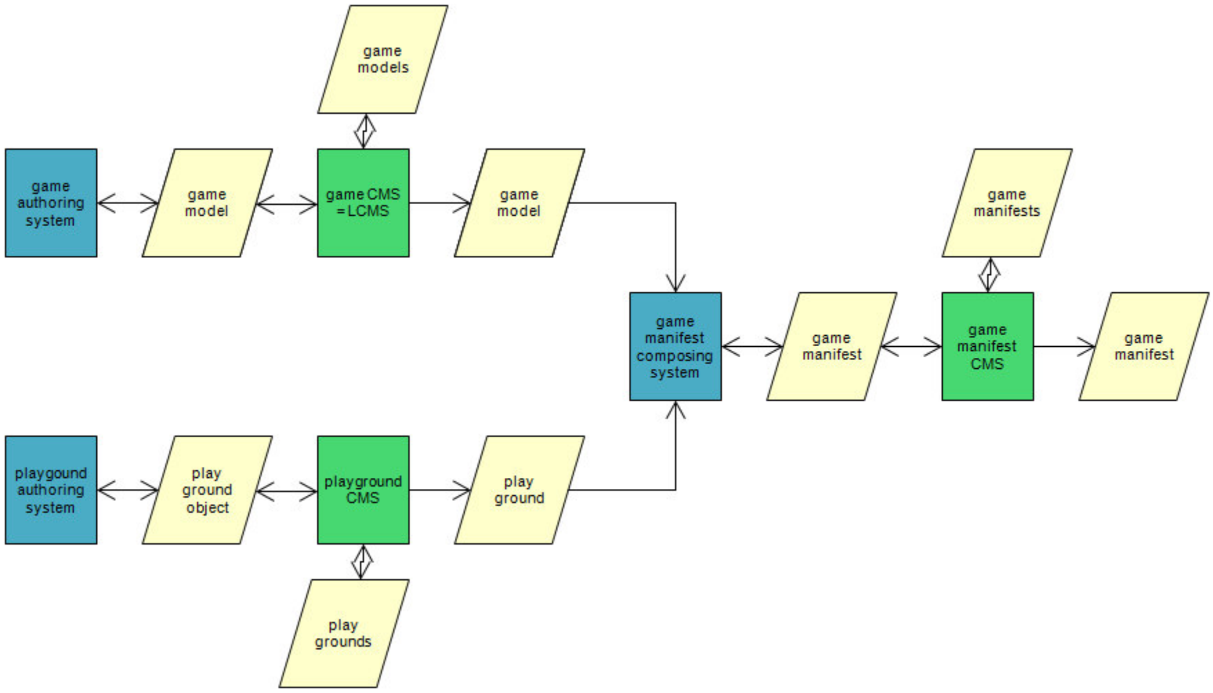


To improve readability we will split up the whole into three pieces:

- authoring and managing content
- (managing) the learning process,
- managing the system (not shown in the figure).

3.3.1 Authoring and managing content

In this paragraph we will deal with the subsystems for authoring content and managing content.



The sub-system for authoring content will be split-up into three sub- subsystems:

- playground authoring system - an interactive environment that enables playground authors to create and edit playground objects, i.e. organisations or individuals that will appear on the interactive

Cyberdam city map and/or City Directory (old name: Yellow pages) and underlying web sites. The playground is common, i.e. it can be used by a variety of games.

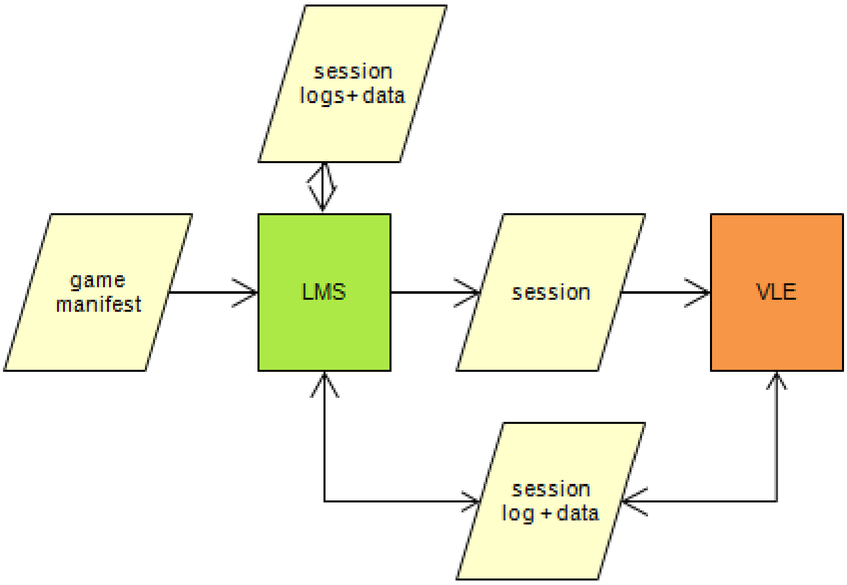
- game authoring system - an interactive environment that enables game authors to build games. These games (as products, also called game- models) are independent of the playground, i.e. at least in theory they can be linked in various ways to a certain playground or (in future versions) can be linked with a variety of playgrounds (or no playground at all).
- game manifest composing system - an interactive environment that enables game manifest composers to compose a game manifest, i.e. a link between a certain game and a certain playground, more specifically between the roles of the game and the objects of the playground.

The sub-system for managing content, the equivalent of a Learning Content Management System, will be split-up into three sub-sub- systems as well:

- playground content management system - an interactive environment that enables playground authors and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) playground objects.
- game content management system - an interactive environment that enables game authors and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) games.
- game manifest content management system - an interactive environment that enables game manifest composers and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) game manifests.

3.3.2 (managing) the learning process

In this paragraph we will deal with the sub-systems for managing the learning process (i.e. the game sessions) and the learning process itself.



The sub-system for managing the learning, the equivalent of a Learning Management System, is an interactive environment that enables game masters to register/enrol participants, and to start, administer and stop game sessions based on a game manifest. In future versions it enables them to manage session logs + data, to archive them (as a report or as a file) and to remove them from the system. Game masters are restricted to their own sessions; the LMS administrator has access to all.

The sub-system for learning, the equivalent of a Virtual Learning Environment, enables game (session) participants (students, tutors) to participate to a game session, i.e. play a certain role. It further enables game masters to monitor sessions, even after they have been completed. Game masters can intervene during sessions by changing the role-allocation of participants and by taking over a role for some time. Game masters are restricted to their own sessions; the LMS administrator has access to all.

3.3.3 Managing the system

Managing the system (not shown in figure) is an interactive environment that enables all system users to inspect and/or change personal settings, like dialogue language and notification preferences. It enables system administrators to register users and their privileges, to maintain a variety of dialogue languages, to maintain the interactive city map, and to manage all data involved. Automatically it does some housekeeping like sending out e-mail notifications.

3.4 Data view

The figures in the previous section describe the relation between the Cyberdam sub- systems and the data flows involved.

Cyberdam 2.0 works with the following data objects:

- Playground object - basic data regarding organisations or individuals that will appear on the interactive Cyberdam city map and/or City Directory (old name: Yellow pages) and underlying web sites. It consists of name, category, geographical coordinates, popup text, thumbnail, picture, description, and optional web pages. Owner of a playground object is the user who created or copied it, usually a playground object editor.
- Playground - an interactive city map plus corresponding playground objects.
- Playgrounds - more than one playground (Note: Cyberdam 2.0 has only one playground, the virtual city of Cyberdam).
- Game model (or game) - metadata, a gameplay description (earlier: workflow), a list of roles, activity descriptions and related multimedia documents. Owner of a game is the user who created or copied it, usually a game author.
- Game models (or games) - a collection of games.
- Game manifest - a named link between a certain game and a certain playground, more specifically between the roles of the game and the objects of the playground. Owner of a game manifest is the user who created or copied it, usually a game manifest composer.
- Game manifests - a collection of manifests.
- Game session - an instance of a game, based on a game manifest (that refers to a game and a playground) and participants allocated to certain roles. Owner of a game session is the user who created or copied it, usually a game master.
- Session log + data - the logging of a game session and the data files produced (e.g. uploaded) during the game session. Owner of a session log + data is the owner of the corresponding session.
- User settings - (not shown in figure) the more-or-less fixed data stored for each user are called the user profile, .personal data on users, the groups to which they belong, the access rights of these groups, and personal settings like the preferred dialogue language. Owner of the user settings is the user to whom they refer (although some settings cannot be changed freely by the user himself). Nobody can see a registered user password.
- Dialogue texts - (not shown in figure) text strings for multi-language support. Owner of the dialogue texts is the system administrator.
- System parameters - (not shown in figure) parameter values used by the system. Owner of the system parameters is the system administrator.

3.5 Users view

Cyberdam 2.0 makes a distinction between the following groups of users with associated access rights:

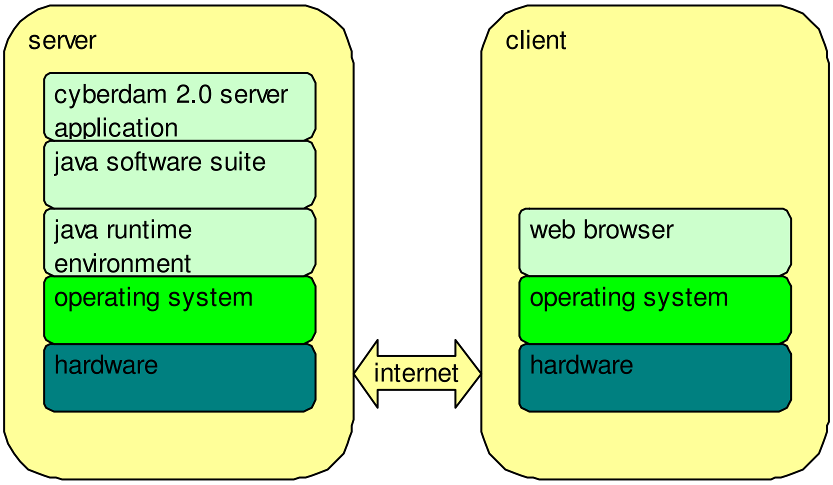
- General Public - Users that have not logged in have no access at all to the system.
- Participant - Every registered user may participate in one or more game sessions, i.e. play one or even more roles in such session(s). A typical participant will be a student. However, also teaching staff and/or external experts may act as participants. A role can be played by a single participant or by more participants, operating as a team. If a role is played by a team, each team member has access to the session.

  During a game session a participant can:

  - access the interactive city map and the directory of Cyberdam, the underlying information and web pages,
  - send messages to and receive messages from other roles in the same session,
  - store and retrieve files to/from the file directory belonging to the own role,
  - inspect a list of pending activities (depending on the role and the stage of the session) and execute these activities (again sending messages and storing/uploading files, but also progress activities); note that completion of a progress activity may bring the session in another stage (with other activities enabled).
- Game session master (old name Case Author) - A Game Session Master, or Game Master for short, may select a game manifest, enrol users as participants and start a game session. During the session, the Game Master can monitor its progress and intervene in a limited way. If more elaborate interventions are desirable, the game author may build a dedicated role into the game itself. The Game Master may allocate this role to a tutor. A Game Master can only monitor her own sessions.
- Game manifest composer - A game manifest composer establishes a link between an existing game and an existing playground. She links roles to playground objects. This results in a game manifest, a Game Master needs to start a game session.
- Game Author (old name Template Author) - A game author creates a game consisting of metadata, roles, activities, resources and a game play connecting them. A game author may publish his game to be used by other authors or to be used by game manifest composers. If shared with other authors, they may make a private copy of it and use it as starting point for a new game. A game author can only change and/or delete his own games.
- Playground author - The Playground Author can create/change playground objects (i.e. city characters). Playground authors can change each others objects.
- LCMS administrator - The LCMS administrator manages the games, playground objects and game manifests.
- LMS administrator - The LMS administrator has access to the LMS functions of all sessions.
- VLE administrator - The LMS administrator has access to the VLE functions of all sessions.
- User administrator - The User Administrator registers new users, updates their data and removes them when necessary.
- System Administrator - The System Administrator registers users and privileges, maintains the multi-language text strings and system parameters.

3.6 Technical view

In technical sense Cyberdam 2.0 is a client-server system. Client and server subsystems are connected via the Internet.



The server subsystem consists of a stack of software packages. At the top of it run the Cyberdam 2.0 components. The next lower level is taken by a Java suite (see table). The following levels contain the Java Runtime Environment and the Operating System.

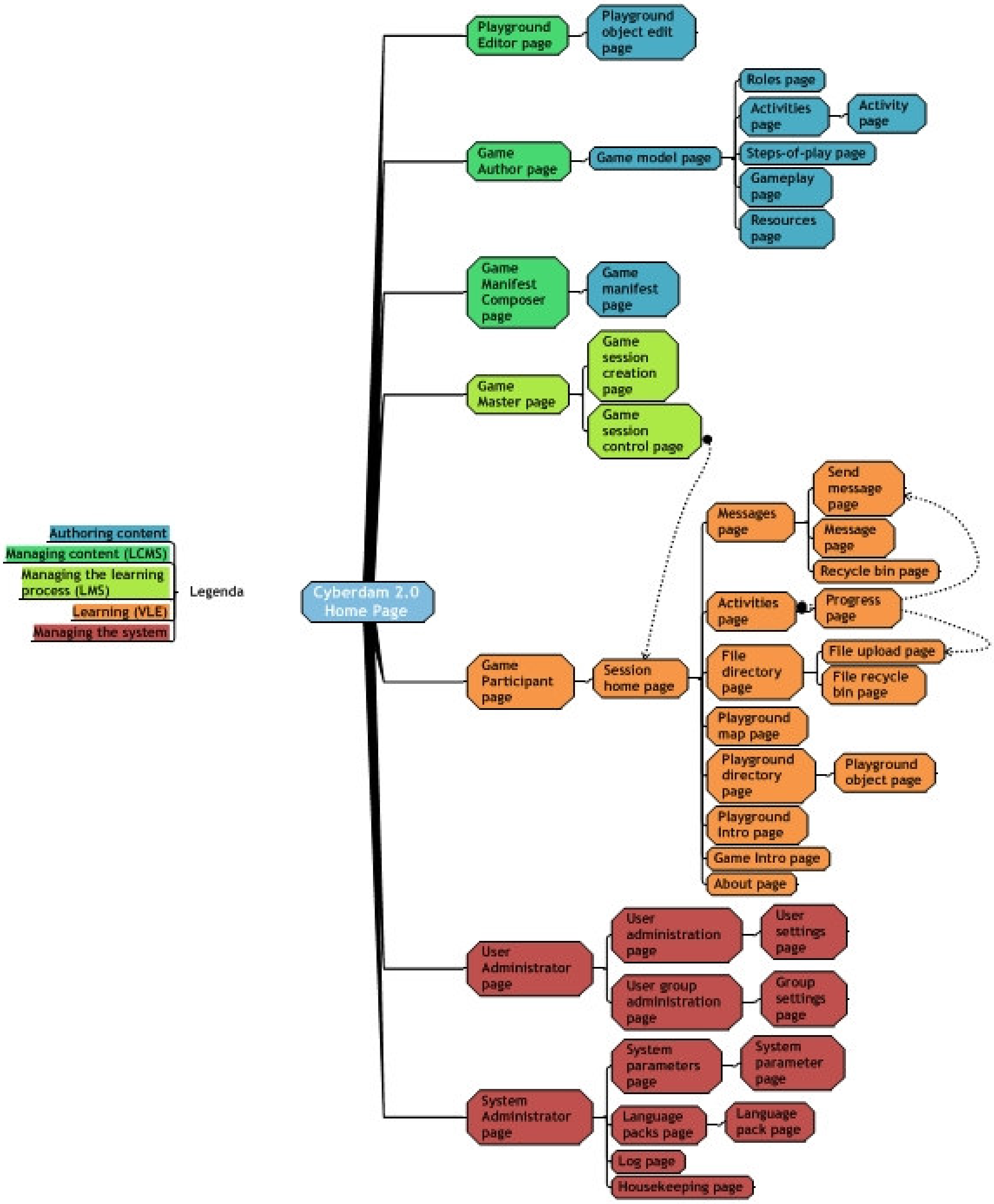
Name	**Function**
Apache Tomcat	Servlet container
Hibernate	Object-Relational mapping
Log4J	Logging
MySQL	Database management
Open AMF	Flash remote component
Spring framework	MVC framework

The client subsystem is nothing more than a device running a web browser. Although in general, Cyberdam 2.0 could work with arbitrary web browsing devices, the user interface will be based on regular PC-screen sizes and the most popular browsers like Mozilla FireFox and MS Internet Explorer with an up-to-date Flash plugin.

4 Site map

The users invoke the Cyberdam functions through web pages. This chapter describes these web pages from a user point of view. It starts with the Cyberdam home page and continues with the pages accessible from the Cyberdam home page. The latter pages may have subpages as well. Ultimately the chapter refers to the very system functions to be invoked. The latter will be described in the next chapter.

The figure below shows the pages and their relations. The colors indicate the subsystems as used in the earlier sections on Business view and Systems view.



4.1 Home page

Access:

The home page is accessible for everybody.

Purpose:

- Force the user to login
- Give the logged in user an overview and a pathway to the pages relevant for him.

Content:

As long as the user has not been logged in, the Home page displays:

- title of the page, e.g. "Welcome to Cyberdam"
- system version (see System Variables)
- copyright notice (see System Variables)
- introduction text, e.g. "Please, login"
- Login name entryfield
- Password entry field (input will not be displayed)
- Login button

Once logged in, the Home page displays:

- title of the page, e.g. "Welcome to Cyberdam"
- system version (see System Variables)
- copyright notice (see System Variables)
- introduction text
- a link to the Game Participant page (for all users)
- if the function SystemManagement.User.Access returns OK for Game Masters, VLE Administrators, LMS Administrators and System Administrators, a link to the Game Master page is displayed
- same for Game Manifest Composers, LCMS Administrators and System Administrators: a link to the Game Manifest Composer page
- same for Game Authors, LCMS Administrators and System Administrators): a link to the Game Author page
- same for Playground Editors, LCMS Administrators and System Administrators: a link to the Playground Editor page
- same for User Administrators and System Administrators: a link to the User Administrator page
- same for System Administrators: a link to the System Administrator page
- a link to the Personal settings page (for all users)
- a link to the SystemManagement.User.Logout function (for all users).

Once logged out, the Home page is redisplayed again (see above).

Actions:

As long as the user did not login:

- The user can fill the entryfields.
- Clicking the Login button calls the SystemManagement.User.Login function.
    - If it returns OK, the user is logged in, and the home page refreshes (showing the home page for logged in users),
    - If it does not return OK, the system displays an error message (e.g. "Your loginname and/or password do not match our records. Please try again or contact your help desk") and the user stays on the current page.

Once logged in, clicking on one of the links, will display the indicated page.

4.2 Playground Editor page

Access:

The Playground Editor page is accessible for Playground Editors, LCMS Administrators and System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the playground editor an overview of all the playground objects (by him and by others) and let him select one to copy and/or change or let him create a new one.

Content:

The Playground Editor page displays:

- title of the page
- introduction text
- table with playground objects
- Delete button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user)
- Edit button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user)
- Copy button (for every row)
- Close button

Table:

The table with playground objects has the following columns:

URI, e.g. "sd- 012"	object name	object category	map and/or directory? (Y/N Y/N)	status	object version (i.e. last mutated) (date + time)	owner, i.e. playground editor (full name)

- The default order is: object name.
- Filter: for LCMS administrators and system administrators all objects are shown; for other users the own objects are shown plus the objects owned by others as long as they have status = 3 = public.

Actions:

- Clicking the Edit button for a certain object will display the Playground object edit page for that object.
- Clicking the Delete button for a certain object will display a dialogue box for confirmation. If confirmed the object will be deleted.
- Clicking the copy button will call the LCMS.PlaygroundObject.Copy function which copies the playground object and makes the current user owner of the new object.
- Clicking the New Playground Object button will display an empty Playground object edit page.
- Clicking on the Close button brings the user to the Home page.

4.2.1 Playground object edit page

Access:

The Playground object edit page is accessible for Playground Editors, LCMS Administrators and System Administrators. It can be reached via a link on the Playground Editor page.

Purpose:

Let the playground editor create or edit a playground object.

Content:

The Playground object edit page displays:

- title of the page
- introduction text
- a form (either filled or empty) with the following fields:
    1. URI, e.g. "sd012"
    2. object name
    3. object caption (max. 127 chars)
    4. object category
    5. object thumbnail (picture to be used in popup window on city map)
    6. object picture (picture to be used in Playground object page)
    7. object map coordinates (x, y)
    8. created by
    9. created on
    10. last mutated by
    11. last mutated on
    12. visible on map? (Y/N, default = N)
    13. listed in directory (Y/N, default = N)
    14. status (default = 1 = under construction)
- OK button
- Cancel button

Actions:

- The user can enter new data in the entryfields of the form or overwrite existing data.
- Clicking the OK button results in checking the data, displaying eventual warnings, saving the data, and returning to the Playground Editor page.
- Clicking the Cancel button results in abandoning the changes and returning to the Playground Editor page.

4.3 Game Author page

Access:

The Game Author page is accessible for Game Authors, LCMS Administrators and System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the game author an overview of all his games and the games made available by other authors and let him select one to copy and/or change or let him create a new one.

Content:

The Game Author page displays:

- title of the page
- introduction text
- table with all the games owned by the user and all the games authored by other users with status = 3 = public. If the user is an LCMS Administrator or System Administrator all games of all users are displayed.
- Edit button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user).
- Delete button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user)
- Copy button (for every row)
- New game button.
- Close button

Table:

The table with games has the following columns:

game name	status	# of roles	description		game version (= last mutation)(date+time)	owner, i.e. game author (full name)

The default order is: game author + game name.

Actions:

- Clicking the Edit button results in calling LCMS.Game.Edit and displaying the Game model page for that game.
- Clicking the Delete button will result in a request for confirmation followed by the deletion of the game. This will be logged by SystemManagement.Log.Add.
- Clicking the Copy button will call the LCMS.Game.Copy function which copies the game and makes the current user owner of the new game.
- Clicking the New Game button will display an empty Game model page.
- Clicking on the Close button brings the user to the Home page.

4.3.1 Game model page

Access:

The Game model page is accessible for the Game Author who owns the game, LCMS Administrators and System Administrators. It can be reached via a link on the Game Author page.

Purpose:

Let the game author create or change a game.

Content:

The Game model page contains the Game model wizard with the following secondary pages (final saving occurs on the last page):

- Introduction page
- Metadata page
- Roles page
- Activities page
- Steps-of-play page
- Gameplay page (earlier called workflow)
- Resources page.

Initially the wizard displays the Introduction page (see there).

Actions:

- See the secondary pages for more information.

Introduction page

Access:

The Introduction page is part of the wizard on the Game model page.

Purpose:

Give the game author a short introduction on the wizard.

Content:

The Introduction page displays:

- title of the page
- introduction text
- Next button
- Back button (dimmed and disabled)
- Cancel button

Actions:

- Clicking the Next button displays the Metadata page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Metadata page

Access:

The Metadata page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author specify the metadata of the game.

Content:

The Metadata page displays:

- title of the page
- introduction text
- the following entryfields:
  - name of the game
  - description of the game
  - # of roles (calculated field)
  - status of the game (default = 1 = under construction)
- Next button
- Back button (dimmed and disabled)
- Cancel buttom

Actions:

- The status of a game can be: 1 = under construction, 2 = private, 3 = public, 4 = obsolete). Once a game has status 3 = public or status = 4 = obsolete, only a LCMS Administrator or a System Administrator can change the status again.
- Clicking the Next button displays the Roles page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Roles page

Access:

The Roles page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author specify roles and their presentation sequence.

Content:

The Role page displays:

- title of the page
- introduction text
- A table to enter roles.
- A mechanism to change the sequence of the roles.
- Edit button (on each row)
- Delete button (on each row)

- Copy button (on each row)
- New role button
- Next button
- Back button
- Cancel button

Table:

The table with roles has the following columns:

sequence number	role					

The default order is: sequence number.

Actions:

- Clicking the Delete button will:
  - check if the role is used on the other pages of this wizard,
  - if so, refuse to delete the role and give a proper warning message,
  - if not, delete the role.
- Clicking the Copy button will copy a role, the role name is changed to "Copy of"(language dependent) and the role as used in the source row. No related activities are copied.
- Clicking the New role button, will add a new role to the table. There is no maximum number of roles.
- Manipulating the sequencing mechanism, will change the sequence of the roles. This affects only the sequence as displayed to the participants.
- Clicking the Next button displays the Activities page.
- Clicking the Back button displays the Metadata page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Activities page

Access:

The Activities page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author specify activities.

Content:

The Activities page displays:

- title of the page
- introduction text
- A table with activities.
- Edit an activity button (on each row)
- Delete an activity button (on each row)

- Copy an activity button (on each row)
- New activity button
- Cancel button

Table:

The table with activities has the following columns:

Type	Name	Status				

The default order is: type+name.

Actions:

- Clicking the Edit activity button opens an Activity page in a popup screen with the activity data filled in.
- Clicking the Delete button will:
  - check if the activity is used on the GamePlay page of this wizard,
  - if so, refuse to delete the activity and give a proper warning message,
  - if not, delete the activity.
- Clicking the Copy button activity copies the activity. The new activity is renamed to "Copy of" (language dependent) followed by the name of the copied activity.
- Clicking the New activity button adds a row/role to the table, type and name have to be filled in.
- Clicking the Back button displays the Roles page.
- Clicking the Next button displays the Steps-of-play page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Activity page

Access:

The Activity page is accessible from the Game model wizard on the Game model page.

Purpose:

Let the game author specify an activity.

Content:

The Activity page displays:

- title of the page
- introduction text
- a form, its entryfields depend on the activity type (message, file upload, progress);
  - for all types:
    1. title of the page
    2. introduction text
    3. name of the activity
    4. instruction field

        5. list of attachments
    - for Message-type activity:
        - default text for the body of the message
        - list of recipients (=roles) for the message.
    - for File upload type activity:
        - no extra fields
    - for Progress type activity:
        - list of options (max. 15)
            - description
            - follow-up action (symbolic name of the step of play)
        - default option (0-15)
    - OK button
    - Cancel button

Actions:

- The user can enter new data in the entryfields of the form or overwrite existing data.
- Clicking the OK button results in checking the data, displaying eventual warnings, saving the data, and returning to the Activities page.
- Clicking the Cancel button results in abandoning the changes and returning to the Activities page.

Steps-of-play page

Access:

The Steps-of-play page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author specify steps-of-play.

Content:

The Steps-of-play page displays:

- title of the page
- introduction text
- A table to enter steps-of-play, i.e. the sequence of the steps as displayed on the gameplay page.
- A mechanism to change the sequence of the roles.
- Edit button (on each row)
- Delete button (on each row)
- Copy button (on each row)
- New step-of-play button
- Back button
- Next button
- Cancel button

Table:

The table with roles has the following columns:

sequence number	symbolic name of the step (by default step-001 eyc.)	name of the step				

The default order is: sequence number.

Actions:

- Clicking the Edit button results in opening the selected row for editing. Step sequence numbers, names and symbolic names may be changed.
- Manipulating the sequencing mechanism, will change the sequence of the steps. This affects only the sequence as displayed to the game author on the Gameplay page (see below).
- Clicking the Delete button will:
  1. check if the step has a related activity on the Gameplay page of this wizard,
  2. if so, refuse to delete the step and give a proper warning message,
  3. if not, delete the step.
- Clicking the Copy button will copy a step; the step name is changed to "Copy of"(language dependent) plus the step name as used in the source row. No activities from the Gameplay will be copied.
- Clicking the New step button, will add a new step to the table. There is no maximum number of steps.
- Clicking the Back button displays the Activities page.
- Clicking the Next button displays the Gameplay page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Gameplay page

Access:

The Gameplay page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author specify the gameplay (earlier called workflow).

Content:

The Gameplay page displays:

- title of the page
- introduction text
- A matrix of roles (rows) and steps-of- play (colums).
- Add new activity button
- Delete an activity button (for each activity in the matrix)
- Back button

- Next button
- Cancel button

Actions:

- Clicking on a cell makes the cell the current cell.
- Clicking the Add an activity button opens a browse list of activities, selecting one of them places the activity in the current cell of the matrix. Note that more than one activities in one cell are allowed. Note that a single activity may appear in a number of cells.
- Clicking the Delete an activity button deletes the selected activity from the current matrix cell.
- Clicking the Back button displays the Steps-of-play page.
- Clicking the Next button displays the Resources page.
- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

Resources page

Access:

The Resources page is part of the Game model wizard on the Game model page.

Purpose:

Let the game author upload and manage resources to be used during the game. e.g. as attachments to game activities.

Content:

The Resources page displays:

1. title of the page
2. introduction text
3. A table with resources and their names, i.e. the names as displayed during the game.
4. Edit button (on each row)
5. Delete button (on each row)
6. Browse button with entryfield
7. New resource button
8. Back button
9. Finish button
10. Cancel button

Table:

The table with resources has the following columns:

name	file name (read only)	file size (calculated field)	last mutated on (date+time)	last mutated by (full name)	created on (date+time)	created by (full name)

The default order is: name.

Actions:

- Clicking the Edit button results in opening the selected resource entry for editing. The resource name and the comment fields may be changed.
- Clicking the Delete button will delete the resource entry plus the resource file itself.
- Clicking the Browse button will open a popup window to search for files on the local system. If a file has been selected, its URL is ebtered in the entryfield automatically.
- Clicking the New resource button, will add a new resource to the table and upload the file selected via the Browse button. There is no limit to the number of resources.
- Clicking the Back button displays the Gameplay page.
- Clicking the Finish button results in:

  1. calling LCMS.Game.Save to check the game and give some warnings if appropriate
  2. if the check returns OK, the game is saved and the user returns to the Game author page, otherwise he stays on the current Game model page.

- Clicking the Cancel button abandons any changes on all the wizard's pages and displays the Game Author page again.

4.4 Game Manifest Composer page

Access:

The Game Manifest Composer page is accessible for all Game Manifest Composers, LCMS Administrators and System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the game manifest composer an overview of existing game manifests and let him select one to copy and/or change or let him create a new one.

Content:

The Game Manifest Composer page displays:

- title of the page
- introduction text
- table with all the game manifests owned by the user and the manifests made available by other authors (or ALL game manifests of all users if the user is an LCMS Administrator or a System Administrator)
- Edit button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user)
- Delete button (for every row if the user is LCMS administrator or system administrator, otherwise only for rows owned by the user)
- Copy button (for every row)
- New manifest button
- Close button

Table:

The table with game manifests has the following columns:

manifest name	game name	playground name	# of roles	status	version, i.e. last mutated (date+time)	owner, i.e. manifest composer (full name)

The default order is: manifest name (ascending).

Actions:

- Clicking the Edit button will display the Game manifest page of that manifest.
- Clicking the Delete button will result in a request for confirmation followed by the deletion of the manifest. This will be logged by SystemManagement.Log.Add.
- Clicking the Copy button will call the LCMS.GameManifest.Copy function which copies the game manifest and makes the current user owner of the new one.
- Clicking the New Manifest button will display an empty Game manifest page.
- Clicking on the Close button brings the user to the Home page.

4.4.1 Game manifest page

Access:

The Game manifest page is accessible for all Game Manifest Composers who created it, LCMS Administrators and System Administrators. It can be reached via a link on the Game Manifest Composer page.

Purpose:

Let the game manifest composer create or change a game manifest.

Content:

The Game manifest page displays:

- title of the page
- introduction text
- a form with the following entryfields:
  - manifest name
  - game name (plus Browse button)
  - playground name (for the moment "Cyberdam")
  - # of roles (calculated field, derived from game)
  - status
  - a table with roles related to playground objects plus a Browse button
- OK button
- Cancel button

Table:

The table with roles-playground objects has the following columns:

role name	playground object name	playground object category	playground object caption	playground object URI		

The default order is: role name.

Actions:

- Clicking on the Browse button for game names displays a popup screen with a table of available games (own games with status = 3 = private or all games with status = 4 = public) plus a Close button. Actions on the popup screen:
  1. Selecting a game results in writing its name in the game name entryfield on the Create/edit manifest form and closing the popup- screen.
  2. Clicking the Close button will close popup screen.
- As long as no game has been specified, the table with roles- playground objects is empty. As soon as a game name has been specified, the table will be filled with the roles of that game.
- Clicking on the Browse button for a row in the table, displays a popup screen with a table of available playground objects plus a Close button. Actions on the popup screen:
  1. Selecting a playground object results in writing its URI, name, category and caption in the corresponding table fields and closing the popup- screen.
  2. Clicking the Close button will close popup screen.

- Clicking OK results in creating or updating the manifest and returning to the Game Manifest Composer page.
- Clicking Cancel results in abondoning any changes and returning to the Game Manifest Composer page.

4.5 Game Master page

Access:

The Game Master page is accessible for Game Masters, VLE Administrators, LMS Administrators and System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the game master an overview of all the game sessions he is in charge of and let him select one to monitor or intervene or let him create a new one.

Content:

The Game Master page displays:

- title of the page
- introduction text
- table with all the game sessions in which the user is or has been acting as Game Master (or ALL sessions if the user is a VLE Administrator, an LMS Administrator or a System Administrator).
- Edit button (for every row if the user is LMS administrator, VLE administrator or system administrator, otherwise only for rows owned by the user)
- Delete button (for every row if the user is LMS administrator, VLE administrator or system administrator, otherwise only for rows owned by the user)
- Copy button (for every row)
- New session button
- Close button.

Table:

The table with game sessions has the following columns:

session name	status	actual status started (date + time)	# of roles	# of participants	step-of- play, activities of all roles (uncompleted/total)	owner, i.e. game master (full name)

The default order is: status (ascending) + running started (descending).

Actions:

- Clicking the Edit button will display the Game session control page for that session.
- Clicking the Delete button will result in a request for confirmation followed by the change of the session status from status = 1 or status = 2 to status = 6, or from status = 3 to status = 5; status =5 cannot be "deleted". This will be logged by SystemManagement.Log.Add.

1. Clicking the Copy button will call LMS.Session.Copy which results in a new game session with status = 1 = created.
2. Clicking the New Session button will display the Game session creation page.

- Clicking the Close button will display the Home page.

4.5.1 Game session creation page

Access:

The Game session creation page is accessible for Game Masters, VLE Administrators, LMS Administrators and System Administrators. It can be reached via a button on the Game Master page.

Purpose:

Let a game master create a new game session.

Content:

The Game session creation page shows:

1. title of the page
2. introduction text
3. an empty form with the following entry fields:
    - name of the session
    - name of the manifest
    - a Browse button (to search for a manifest)
    - OK button
    - Cancel button

Actions:

- Clicking the Browse button displays a popup screen with a table of available manifests plus a Close button. Actions on the popup screen:
    - Selecting a manifest results in writing its name in the manifest entryfield on the Game session creation page and closing the popup- screen.
    - Clicking the Close button will close the popup screen.
- Clicking the OK button results in creating a new session with status 1 = Created (session has a game master, a name and a manifest) and displaying the Game Master page again.
- Clicking the Cancel button abandons any changes and displays the Game Master page again.

4.5.2 Game session control page

Access:

The Game session control page is accessible for Game Masters, VLE Administrators, LMS Administrators and System Administrators. It can be reached via a link on the Game Master page.

Purpose:

Let the game master start-up and shut-down a game session, monitor it and intervene.

Content:

The Game session control page displays:

- title of the page
- introduction text
- a form with the following fields:
    1. session name (read only)

2. status (read only)
3. actual status started (date+time) (read only)
4. running started (read only)
5. session stopped (read only)
6. a table with all roles - participants (by default all roles have been allocated to the game master) (behaviour of this table depends on the status).
7. a Browse button to search for user names
8. Edit button (on each row)
9. Delete button (on each row)
10. Copy button (on each row)
11. Monitor button (on each row)
12. Control button
13. Close button

Table:

The table with roles-participants has the following columns:

role name	playground character name	user name	first name of user	last name of user		

The default order is: role name + user name.

Actions:

1. When status = 1 = Created
    1. Clicking the Edit button opens a row for editing.
        1. The user can allocate a role, i.e. fill in a user name in the role-participants table. Clicking the Browse button results in a popup screen containing a user table (see below).
        2. The user can de-allocate a role, i.e. clear a user name in the role-participants table. To avoid a role to become vacant, the system will automatically fill in the game master's name in stead of the cleared participant name.
    2. Clicking the Delete button will, after confirmation, result in the deletion of a role-participant row, i.e. decrease the number of participants in a team role. This is only possible as long as there remains at least one participant for this role.
    3. Clicking the Copy button creates a new row for a certain role. In this way the user can duplicate a role (i.e. create a multi- player role or team role); i.e. copy a row in the role-participants table and write the user name of the second participant in the copied row.
    4. The Monitor button is not visible.
    5. Clicking the Control button gives the opportunity to change the session status:
        1. From 1 (Created) to 2 (Ready-to-start) when all roles are filled with participants
        2. From 1 (Created) to 6 (Cancelled) when the session is cancelled before it started.
    6. Any change will result in an entry to be added to the system log via SystemManagement.Log.Add.

2. When status = 2 = Ready to run
    1. The Edit, Delete, Copy and Monitor buttons are not visible.
    2. Clicking the Control button gives the opportunity to change the session status:
        1. From 2 (Ready-to-run) back to 1 (Created) when role allocation needs extra attention, or simply to avoid accidental starting of the session.
        2. From 2 (Ready-to-run) to 3 (Running). From that moment on the session becomes visible for the participants.
    3. Any change will result in an entry to be added to the system log via SystemManagement.Log.Add.
3. When status = 3 = Running
    1. Clicking the Edit button opens a row for editing.
        1. The user can allocate a role, i.e. fill in a user name in the role-participants table. Clicking the Browse button results in a popup screen containing a user table (see below).
        2. The user can de-allocate a role, i.e. clear a user name in the role-participants table. To avoid a role to become vacant, the system will automatically fill in the game master's name in stead of the cleared participant name.
    2. Clicking the Delete button will result, after confirmation, in the deletion of a role-participant row, i.e. decrease the number of participants in a team role. This is only possible as long as there remains at least one participant for this role.
    3. Clicking the Copy button creates a new row for a certain role. In this way the user can duplicate a role (i.e. create a multi- player role or team role); i.e. copy a row in the role-participants table and write the user name of the second participant in the copied row.
    4. Clicking the Monitor button opens the session home page of a certain role.
    5. Clicking the Control button gives the opportunity to change the session status:
        1. From 3 (Running) to 4 (Normal end) is only possible via Progress activities within the game; not via session control.
        2. From 3 (Running) to 5 (Abnormal end)
    6. Any change will result in an entry to be added to the system log via SystemManagement.Log.Add.
4. When status = 4 = Normal End
    1. The Edit, Delete, Copy and Control buttons are not visible.
    2. Clicking the Monitor button opens the session home page of a certain role.
5. When status = 5 = Abnormal end
    1. The Edit, Delete, Copy and Control buttons are not visible.
    2. Clicking the Monitor button opens the session home page of a certain role.
6. Clicking on the Close button brings the user back to the Game master page.

4.6 Game Participant page

Access:

The Game Participant page is accessible for all logged-in users. The information displayed is specific for each user. It can be reached via a link on the Home page.

Purpose:

Give the participant an overview of all his own game sessions and let him select one to play (or to view).

Content:

The Game Participant page displays:

- title of the page
- introduction text
- a table with all the game session roles allocated to the user (or ALL game sessions for ALL users if the user is a VLE Administrator or a System Administrator). If this table contains one session role only, the whole page is skipped and the Session home page (dashboard) of that session for the role played by the user is displayed.
- Session home page button (on each row)
- Close button

Table:

The table with game session roles has the following columns:

session name	status	session start (date+time)	role + character		step-of-play, activities (uncompleted/total)	owner, i.e. game master (full name)

- The default order is: status + session start (descending).
- Filter: Only sessions with status = 3 (Running), 4 (Normal end) or 5 (Abnormal end) are shown.
- Filter: if the user is NOT VLE administrator NOR system administrator, all sessions in which the user is participating (has a role) are displayed; each session role results in a separate row in the table.
- The step-of-play is the the outcome of function call VLE.session.step-of-play.

Actions:

- Clicking the Session home page button of one of the game sessions will display the Session home page (dashboard) of that session for the role played by the user.
- Clicking on the Close button brings the user back to the Home page.

4.6.1 Session home page

Access:

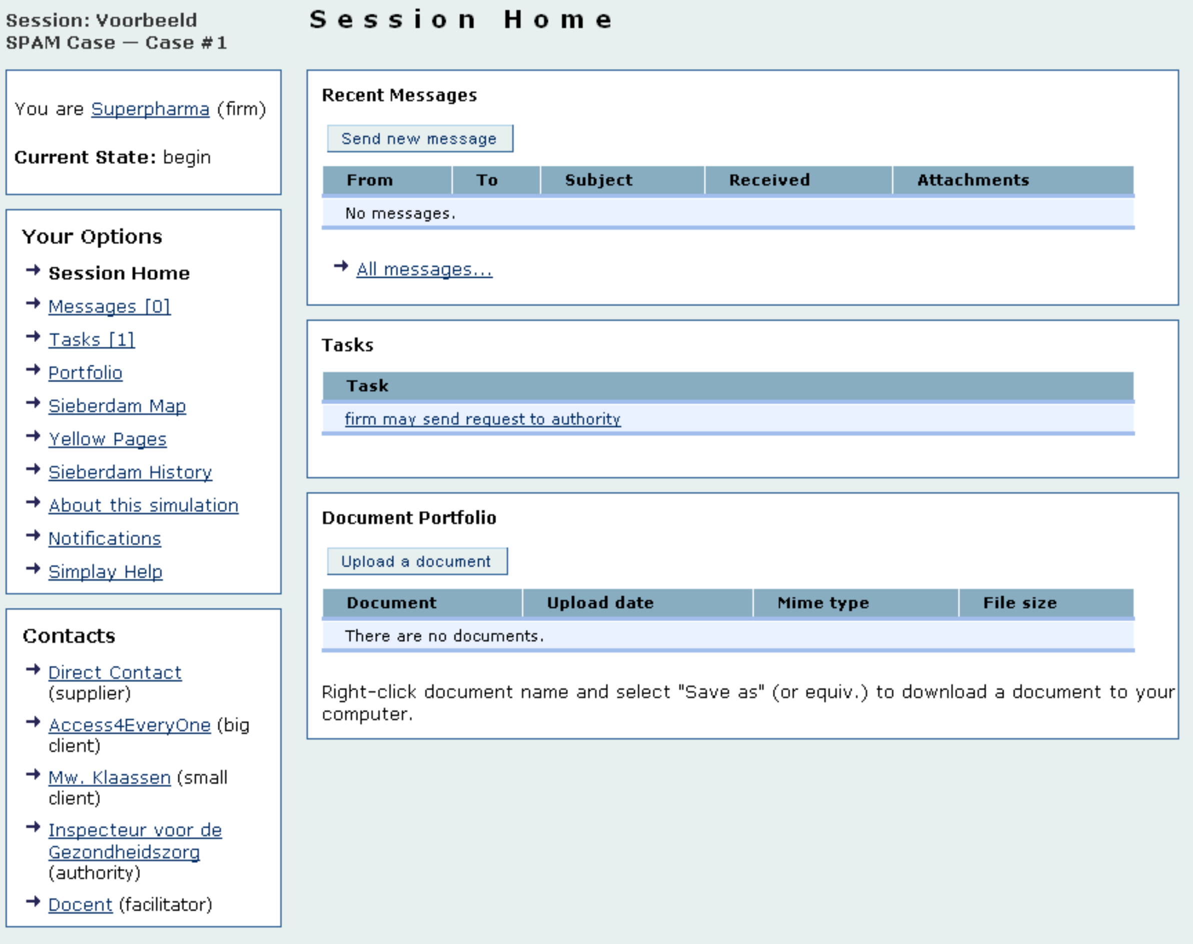
The Session Home page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE administrators, LMS administrators and the system administrator have access to this page. It can be reached via a link on the Game participant page.

Purpose:

Give the participant to a game session a portal page from where all the functions of the game session can be accessed.

Content:

The figure shows a screenshot from Cyberdam 1.1.



Session: Voorbeeld
SPAM Case — Case #1

Session Home

You are Superpharma (firm)

Current State: begin

Your Options

- → **Session Home**
- → Messages [0]
- → Tasks [1]
- → Portfolio
- → Sieberdam Map
- → Yellow Pages
- → Sieberdam History
- → About this simulation
- → Notifications
- → Simplay Help

Contacts

- → Direct Contact (supplier)
- → Access4EveryOne (big client)
- → Mw. Klaassen (small client)
- → Inspecteur voor de Gezondheidszorg (authority)
- → Docent (facilitator)

Recent Messages

Send new message

From	To	Subject	Received	Attachments
No messages.				

→ All messages...

Tasks

Task
firm may send request to authority

Document Portfolio

Upload a document

Document	Upload date	Mime type	File size
There are no documents.			

Right-click document name and select "Save as" (or equiv.) to download a document to your computer.

This home page contains two columns: a navigation column and a work area column. The navigation column consists off:

- Status information (the results of the function VLE.session.status)
- A list of options giving access to:
  - the Messages page
  - the Activities page
  - the File directory page
  - the Playground map page (interactive map)
  - the Playground directory page (earlier called the list of Yellow pages)
  - the Playground Intro page (e.g. the history of Cyberdam)
  - the Game Intro page (e.g. the purpose of the game)

    - the About page (e.g. name, version and author of the game)
    - the Game Participant page
- A list of contacts (i.e. the roles involved in this session plus the related playground objects) giving access to the related Playground object page.
- A toggle to set e-mail notifications for incoming messages On/Off. The default value is found in the user settings.
- A toggle to set e-mail notifications for new steps-of-play On/Off. The default value is found in the user settings.

The work area column consists of:

1. title of the page
2. introduction text
3. The new messages container with recently incoming messages plus a Send New Message button; the maximum number of messages is determined by a system parameter (VLE.MAIL, e.g. 5); this is a stripped version of the work area of the Messages page: the Inbox table and a Send New Message button.
4. The new activities container with the newest activities; the maximum number of activities is determined by a system parameter (VLE.ACTIVITIES, e.g. 5); this is a stripped version of the work area of the Activities page: a table of activities.
5. The new files container with the newest file uploads plus a Browse button and an Upload button; the maximum number of files is determined by a system parameter (VLE.FILES, e.g. 5); this is a stripped version of the work area of the File directory page: a table of files, a Browse button and an Upload button.

Actions:

- Clicking on an element of the navigation column displays the indicated page.
- Clicking an element of the work area has the effect as described on:
    - Messages page
    - Activities page
    - File directory page

4.6.2 Messages page

Access:

The Messages page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) manage incoming and outgoing messages (from and to other roles).

Content:

6. The Messages page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.
7. The work area shows:
    1. title of the page

2. introduction text
3. an inbox, i.e. a table of incoming messages
4. a Delete button (on every row of the inbox)
5. an outbox, i.e. a table of sent messages
6. a Delete button (on every row of the outbox)
7. a Send New Message button,
8. a Recycle bin button.

Table:

The inbox table and the outbox table have the same columns:

from (role)	to (role(s))	subject	step-of- play	date+time	attachments	button

The default order is: date+time (descending).

Actions:

- Clicking the Delete button of an Inbox message or an Outbox message will move that message to the Recycle bin.
- Clicking a message will open that message in a Message page.
- Clicking the Send New Message button, opens a Send message page.
- Clicking the Recycle Bin button will open the Recycle bin page.

Send message page

Access:

- The Send message page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.
- There are three ways to reach this page: from the Messages page, from the Activities page and as a sepomse to an earlier message. In the second case the new message is part of a message-type activity.

Purpose:

Let the participant (as a role) prepare and send a message to another role.

Content:

- The New messages page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.
- The work area shows:
  - title of the page
  - introduction text
  - if the new message is part of a message-type activity: an instruction for the user.
  - a new message form with:
    - From field with the name of the sender (role+related playground object) already filled in; this cannot be changed,

        - To field where receivers (role+related playground object) can be filled in (e.g. via check boxes for all the roles)
        - Subject field (initially empty)
        - Body field (initially empty; if the new message is part of a message-type activity, the body field is filled with the default text for the body)
        - Attachment field where file names can be filled in (e.g. via check boxes for all the files from the online file directory)
    - Send button
    - Cancel button

Actions:

- Clicking the Send button will cause the system to check whether To field, Subject field and Content field are empty:
    - if so, the system displays a warning message and returns to the current page;
    - if not
        - the message is sent;
        - a notification message is generated for all users allocated to the receiving roles an aligned with their personal e-mail notification settings for this session;
        - it wil appear in the Inbox of the receiver(s) and in the Outbox of the sender;
        - depending on their notification settings a regular e- mail is sent to the receivers as well;
        - an entry is added to the system log via SystemManagement.Log.Add;
        - if the new message is part of a message-type activity the activity is set to "completed" and the user returns to the Activities page,
        - if not, the user returns to the Messages page.
- Clicking the Cancel button, will cause the system to abandon the new message and to return to either page where it came from, the Messages page or the Activities page.

Message page

Access:

The Message page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) read a message.

Content:

- The Message page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.
- The work area shows:
- title of the page
- introduction text

- a message containing:
    1. From field with the name of the sender (role),
    2. To field with the names of the receivers (roles),
    3. Step-of-play field with the name (or #) of the step- of- play during which the message was sent,
    4. Date and time field with the actual date and time the message was sent,
    5. Subject field,
    6. Content field,
    7. Attachment field with links to the attachments.
- Reply button
- Delete button (if the message belongs to the Inbox or the Outbox)
- Undelete button (if the message comes from the Recycle bin)
- Close button

Actions:

- Clicking an Attachment will either open the attachment or display a dialogue box offering the user the option to either save or open the attachment (this depends on the Internet browser settings).
- Clicking the Reply button will open a Send message page with the role as sender and all other roles as receiver of the message, the subject is "Re: "plus the earlier subject, de body contains the earlier message preceded with ">" on every line.
- Clicking the Close button closes the message and returns the user to the Messages page.
- Clicking the Delete button, will cause the system to move the message from the current location (Inbox or Outbox) to the Recycle bin and returns the user to the Messages page.
- Clicking the Undelete button, will cause the system to move the message from the Recycle bin to its original location (Inbox or Outbox) and returns the user to the Recycle bin page.

Recycle bin page

Access:

The Recycle bin page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) manage the recycle bin for messages. The recycle bin is used to clean-up the file space visually. During game sessions files are not deleted.

Content:

The Recycle bin page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text

- the Recycle bin, i.e. a table of deleted messages
- Close button

Table:

The table with messages has the following columns:

from (role)	to (role(s))	subject	step-of- play	date+time	attachments	button

All messages have an Undelete Delete button. The default order is: date+time (descending).

Actions:

- Clicking a message will open that message in a Message page.
- Clicking the Undelete button of a message will move that message from the Recycle bin back to the original Inbox or Outbox.
- Clicking the Close button opens the Messages page.

4.6.3 Activities page

Access:

The Activities page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) manage game session activities.

Content:

The Activities page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- a table of activities (clickable)
- Activity button (on each row)
- Close button.

Table:

The table with activities has the following columns:

Type	Name	Completed (Y/N)				

The default order is: type+name.

Actions:

1. Clicking an Activity button will open the activity page corresponding with the activity type:
    1. a message type activity will open a Send message page,
    2. a file directory activity will open a File upload page,

    3. a progress type activity will open a Progress page.
2. Clicking the Close button returns the user to the Session home page.

Progress page

Access:

The Progress page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page. The Progress page is part of a progress-type activity.

Purpose:

Let the participant (as a role) take a decision (i.e. execute a progress-type activity).

Content:

The Progress page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

3. title of the page
4. introduction text
5. instruction for the user
6. a series of Option buttons (also known as radio buttons) followed by descriptions of the options (if there is a default option, that will be selected already)
7. OK button
8. Cancel button.

Actions:

- Clicking an Option button will select that option.
- Clicking the OK button (only possible if an option has been selected) will execute the option and
    - changes the step-of-play;
    - if the new step-of-play does not contain any activities for any role the game session will terminate (normal end); the game session status changes from 3 (Running) to 4 (Normal end);
    - a notification message is generated for all users in all roles according to their personal notification settings;
    - the progress activity is set to completed;
    - an entry is added to the system log via SystemManagement.Log.Add;
    - the Activities page is displayed.
- Clicking the Cancel button results in displaying the Activities page.

4.6.4 File directory page

Access:

The File directory page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) manage the game session file directory.

Content:

The File directory page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

9. title of the page
10. introduction text
11. the File directory, i.e. a table of uploaded files
12. Open button (on every row of the files table)
13. Delete button (on every row of the files table)
14. Rename button (on every row of the files table)
15. New file upload button
16. File recycle bin button
17. Close button

Table:

The table with files has the following columns:

file name			step-of- play	date+time		button

Actions:

- Clicking the Open button will either open the file or display a dialogue box offering the participant the option to save or open the file (this depends on the Internet browser settings of the user).
- Clicking the Delete button will move the file to the File recycle Bin.
- Clicking the Rename button will open an entryfield to change the file name.
- Clicking the New file upload button results in displaying the File upload page.
- Clicking the File recycle button displays the File recycle bin page.
- Clicking the Close button displays the Session Home page.

File upload page

Access:

The File upload page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page. There are two ways to reach this page: from the File directory page and from the Activities page. In the latter case the new message is part of a file-upload type activity.

Purpose:

Let the participant (as a role) upload a file to the file directory.

Content:

The File upload page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- if the New file upload is part of file- upload type activity: an instruction for the user,
- Browse button plus entry field,
- Upload button,
- Close button.

Actions:

- Clicking the Browse button will open a browse popup window to select a file on the local hard disk.
- Clicking the Upload function (only available if a file has been selected with the Browse button):
  - the selected file will be uploaded to the File directory;
  - an entry is added to the system log via SystemManagement.Log.Add;
  - if the file upload is part of a file upload activity, the activity is set to be completed and the user returns to the Activities page;
  - if not, the user returns to the File directory page.
- Clicking the Close button, will cause the system to return to the page where it came from, either the File directory page or the Activities page.

File recycle bin page

Access:

The File recycle bin page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) manage the recycle bin for files.

Content:

The File recycle bin page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

1. title of the page
2. introduction text
3. the Recycle bin, i.e. a table of deleted files
4. Open button (on every row of the files table)
5. Rename button (on every row of the files table)
6. Undelete button (on every row of the files table)
7. Close button

Table:

The table with files has the following columns:

file name			step-of- play	date+time		button

Actions:

- Clicking the Open button will either open the file or display a dialogue box offering the participant the option to save or open the file (this depends on the Internet browser settings of the user).
- Clicking the Undelete button will move the file from the File recycle bin to the file directory.
- Clicking the Rename button will open an entryfield to change the file name.
- Clicking the Close button displays the File directory page.

4.6.5 Playground map page

Access:

The Playground map page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session has access to this page.

Purpose:

Let the participant (as a role) navigate the interactive city map and select city objects.

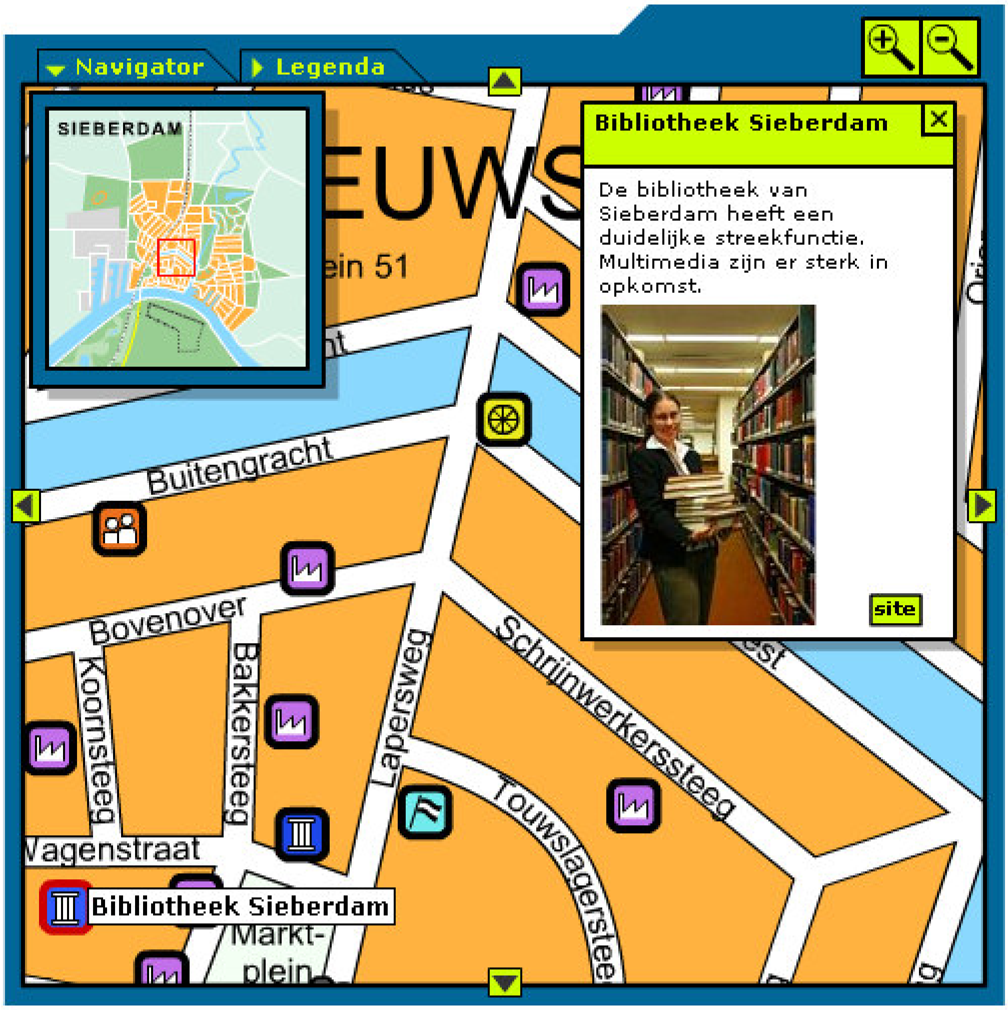
Content:

The Playground map page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

8. title of the page
9. introduction text
10. the common playground, i.e. the interactive map of Cyberdam, with clickable playground objects; which objects are visible and where is determined by the playground object editor.
11. a Close button.

Screenshot:



Bibliotheek Sieberdam

De bibliotheek van Sieberdam heeft een duidelijke streekfunctie. Multimedia zijn er sterk in opkomst.

The figure shows a screenshot of the interactive city map. The screenshot shows an enlarged (zoomed-in) part of the city map with streets, street names and icons. Apparently, the mouse points to an icon (lower left corner). The corresponding city object's name is displayed ("Bibliotheek Sieberdam", the library) and a popup window with a short description, a thumbnail picture and a hyperlink called "site" has been opened (upperright corner). The navigator popup window is open (upper left corner), the legenda popup window is closed. The Navigor Tab and the Legenda Tab act as controls to open and close these popups. The other controls are a zoom-in (+) button, a zoom-out (-) button and arrows in four directions. Not visible is the fact that the key's A and (big) "+" also behave as zoom- in, while the key's "Z" and "-" behave as zoom- out. Outside the map is a hyperlink to the Map Help text.

Actions:

- navigating the map (see screenshot plus description above),
- clicking an object on the map results in the popup of a small window containing a link to the Playground object page for that object.

4.6.6 Playground directory page

Access:

The Playground directory page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Let the participant (as a role) browse the directory of city objects and select city objects.

Content:

The Playground directory page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- a table with playground objects,
- a Close button.

Table:

The tables with playground objects has the following columns:

object name	object category	caption	address			

The default order is: object name.

Playground object page

Access:

The Playground object page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page. There are two ways to reach this page: from the Playground map page and from the Playground directory page.

Purpose:

Show to the participant the content of a playground object page.

Content:

The Playground object page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- Name of the playground object
- Category of the playground object
- Caption
- Address
- Picture

- Description
- Close button

Actions:

- Clicking a hyperlink (if available) in the description field of the playground object opens a new window displaying the page referred to by the hyperlink.
- Clicking the Close button displays either the Playground map page or the Playground directory page.

4.6.7 Playground Intro page

Access:

The Playground Intro is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Give the participant an introduction on the playground.

Content:

The Playground Intro contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- description of the playground (current version: a language dependent text)
- Close button

Actions:

- Clicking the Close button displays the Session home page.

4.6.8 Game Intro page

Access:

The Game Intro is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Give the participant an introduction on the game.

Content:

The Game Intro contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text

- a webpage on the game containing the game description and the splash screen
- a Close button

Actions:

- Clicking the Close button displays the Session home page.

4.6.9 About page

Access:

The About page is a dedicated page for participants playing a certain role in a certain game session. Also the game master of that session, VLE Administrators, LMS Administrators and System Administrators have access to this page.

Purpose:

Give the participant the About Info on the game session.

Content:

The About page contains two columns: a navigation column and a work area column. The navigation column is the same as the navigation column on the Session home page.

The work area shows:

- title of the page
- introduction text
- information on the playground (TBD)
- information on the game (name, caption, created by ... on ..., last mutated by ... on ...)
- information on the manifest (name, caption, created by ... on ..., last mutated by ... on ...)
- information on the session(s) (name, game master)
- Close button

Actions:

- Clicking the Close button displays the Session home page.

4.7 User Administrator page

Access:

The User Administrator page is accessible for User Administrators and System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the user manager an overview of all the user data (entered by him or by others) and let him change these data and register new users.

Content:

The User Administrator page displays:

1. title of the page
2. introduction text
3. link to the User administration page
4. link to the User group administration page
5. a Close button

Actions:

- Clicking on one of the links displays the indicated page.
- Clicking on the Close button displays the Home page.

4.7.1 User administration page

Access:

The User Administration page is accessible for User Administrators and System Administrators.

Purpose:

Let the user administrator inspect/change user data and register new users.

Content:

The User Administration page displays:

- title of the page
- introduction text
- a table with all users
- Edit button (for every row)
- Delete button (on every row)
- New user button
- Browse button to search a file to upload new users
- Upload new users button
- Close button

Table:

The table with users has the following columns:

user-id	full name (given name+family name)	login name	e-mail	language	status	last login (date+time)

- The default order is: family name + given name.
- The default filter is NOT Deleted. An alternative filter is ALL.

Actions:

- Clicking the Edit button results in the display of the User settings page for that user.
- Clicking the Delete button will result in a request for confirmation followed by the change of the status to Deleted. This will be logged by SystemManagement.Log.Add.
- Clicking the Copy button will result in calling SystemManagement.Parameter.Copy to copy the user data.
- Clicking the New User button will display an empty User settings page.
- Clicking the Browse button will open a popup window to select a file on the local hard disk.
- Clicking the Upload new users button will upload and process the selected file; see SystemManagement.User.Upload. It will result in a message with a link to a log file (to be opened or downloaded).

User settings page

Access:

The User settings page is accessible for all users; they see their own settings. It can be reached by clicking their full name which is displayed on every page. User administrators and system administrators can access the User settings pages of all users via a link on the User administration page.

Purpose:

Let the user change some personal settings; let user administrator and system administrator change user settings.

Content:

- The User settings page displays all user settings. They have been subdivided into five categories: basics, privileges, groups, preferences and statistics. It shows:
- title of the page
- introduction text
- basics (visible for the user, changeable by User administrator and System administrator)
  1. given name
  2. family name
  3. login name
  4. password (never visible)
  5. status (enabled, disabled)
- privileges (visible for user, changeable by User administrator and System administrator)
  1. Game session master (Y/N)
  2. Game manifest composer (Y/N)

3. Game Author (Y/N)
4. Playground author (Y/N)
5. LCMS administrator (Y/N)
6. LMS administrator (Y/N)
7. VLE administrator (Y/N)
8. privileges (visible for user, changeable by System administrator)
9. User administrator (Y/N)
10. System Administrator (Y/N)

- groups (visible for user, changeable by User administrator and System administrator)
  - a list of group names the user belongs to
- preferences (changeable by user)
  - dialogue language (default value is SYSTEM.LANGUAGE)
  - e-mail address (used for sending notifications)
  - number of table rows per page (default value is SYSTEM.ROWS)
  - default value for notifications during game sessions
  - message received (Yes/No, default value is VLE.MESSAGE)
  - new step-of-play reached (Yes/No, default value is VLE.STEP)
- statistics (maintained by the system)
- last login (date+time)
- OK button
- Cancel button

Actions:

- Clicking the OK button will save any changes and return the user to the User administration page.
- Clicking the Cancel button will abandon any any changes and return the user to the User administration page.
- Any change in users or user rights is added to the system log via SystemManagement.Log.Add.

4.7.2 User group administration page

Access:

The User group administration page is accessible for User Administrators and System Administrators.

Purpose:

Let the user administrator inspect/change user groups and register new groups.

Content:

The User group administration page displays:

- title of the page
- introduction text

- a table with all user groups,
- Edit button (on every row)
- Delete group button (on every row)
- Copy button (on every row)
- New group button
- Close button.

Table:

The table with users has the following columns:

group category	group name	description	# of members		last mutated (version+date)	last mutated by (full name)

The default order is: group category + group name.

Actions:

- Clicking the Edit button will open the Group settings page for that item.
- Clicking the Delete button will result in a request for confirmation followed by the deletion of the group. This will be logged by SystemManagement.Log.Add.
- Clicking the Copy button will result in calling SystemManagement.UserGroup.Copy to copy the user group.
- Clicking the New group button will open an empty Group settings page.
- Clicking the Close button returns the user to the User Administrator page.

Group settings page

Access:

The Group settings page is accessible for user administrators and system administrators. They access it via a link on the User group administration page.

Purpose:

Let the user change some group settings and inspect/change the members of the group.

Content:

The Group settings page displays:

1. title of the page
2. introduction text
3. entryfields:
    1. group category
    2. group name
    3. description
    4. # of members (readonly)
4. a table with group members
5. Delete member button (every row)

6. Add member button
7. OK button
8. Cancel button

Table:

The table with group members has the following columns:

user-id	full name (given name+family name)	login name	e-mail	language	status	last login (date+time)

The default order is: family name + given name.

Actions:

- Clicking the Delete button will remove a member from the group; the corresponding row wil be deleted from the table.
- Clicking the Add a member button opens a popup screen with all users. Selecting a user will add that user as a member to the group.
- Clicking the OK button will save any changes and display the User group administration page.
- Clicking the Cancel button will abandon any changes and display the User group administration page.

4.8 System Administrator page

Access:

The System Administrator page is accessible for System Administrators. It can be reached via a link on the Home page.

Purpose:

Give the system administrator an overview of the system parameters, the language packs, the system logs and let him change system parameters and language and start some housekeeping functions.

Content:

The System Administrator page displays:

1. title of the page
2. introduction text
3. version of the system (see System Variables)
4. uptime of the system (calculated from System.Start, see System Variables)
5. link to the System parameters page,
6. link to the Language packs page,
7. link to the Log page,
8. link to the Housekeeping page,
9. Close button.

Actions:

- Clicking on one of the links, will display the indicated page.
- Clicking on the Close button brings the user to the Home page.

4.8.1 System parameters page

Access:

The System parameters page is accessible for System Administrators. It can be reached via a link on the System Administrator page.

Purpose:

Give the system administrator an overview of the system parameters and let him change them when appropriate.

Content:

The System parameters page displays:

- title of the page
- introduction text
- a table with all the system parameters
- Edit button (every row)
- Close button

Table:

The table with system parameters has the following columns:

parameter category	parameter name	default value	actual value	comments	version (i.e.last mutated) (date+time)	last mutated by (full name)

- The default order is: parameter category + parameter name.
- The comments field is rendered from the message key in the database table.

Actions:

- Clicking the Edit button for a parameter will display the System parameter page for that parameter.
- Clicking the Close button returns the user to the System Administrator page.

System parameter page

Access:

The System parameter page is accessible for System Administrators. It can be reached via a link on the System parameters page.

Purpose:

Let the system administrator change an existing system parameter value or add a new one.

Content:

The System parameters page displays:

- title of the page
- introduction text
- a form (either filled or empty) with the following fields:
    1. parameter name
    2. initial value (read-only)
    3. actual value
- OK button,
- Cancel button

Actions:

- The user can enter new data in the entryfields of the form or overwrite existing data.
- Clicking the OK button results in checking the data, displaying eventual warnings, saving the data, and returning to the System parameters page.
- Clicking the Cancel button results in abandoning the changes and returning to the System parameters page.

4.8.2 Language packs page

Access:

The Language packs page is accessible for System Administrators. It can be reached via a link on the System Administrator page.

Purpose:

Give the system administrator an overview of the language packs and let him change them when required.

Content:

The Language packs page displays:

- title of the page
- introduction text
- a table with the language packs provided,
- Edit button (on every row)
- Delete button (on every row)
- Copy button (on every row)
- New language pack button
- Close button

Table:

The table with language packs has the following columns:

language code (e.g. "en")	language name					

The default order is: language code.

Actions:

- Clicking the Edit button will display the Language pack page.
- Clicking the Delete button will result in a request for confirmation followed by the deletion of the entry in the language pack list. This will be logged by SystemManagement.Log.Add.
- Clicking the Copy button results in an alert message; in future versions it will result in the copying of a complete language pack.
- Clicking the New Language Pack button results in an alert message; in future versions it will result in the display of an empty Language pack page.
- Clicking the Close button returns the user to the System Administrator page.

Language pack page

Access:

The Language pack page is accessible for System Administrators. It can be reached via a link on the Language packs page.

Purpose:

Let the system administrator inspect and/or change the items of a language pack.

Content:

The Language pack page displays a text file with language texts which can be edited.

Actions:

Possible actions depend on the actual implementation. One implementation could be, opening the text file using the Windows Notepad program. Actions then are editing the text, find and replace, closing the file with or without saving any changes. After closing the text file, the Language packs page is displayed.

4.8.3 Log page

Access:

The Log page is accessible for System Administrators. It can be reached via a link on the System Administrator page.

Purpose:

Give the system administrator an overview of the system logs.

Content:

The Log page displays:

- title of the page
- introduction text
- a table with log entries,
- a Close button

Table:

The table with log entries has the following columns:

module	action	parameter			created on (date+time)	created by (full name)

The default order is: date+time (descending).

Actions:

Clicking the Close button returns the user to the System Administrator page.

4.8.4 Housekeeping page

Access:

The Housekeeping page is accessible for System Administrators. It can be reached via a link on the System Administrator page.

Purpose:

Let the system administrator some housekeeping functions.

Content:

The Housekeeping page displays:

1. title of the page
2. introduction text
3. last housekeeping (date+time)
4. Housekeeping button
5. Close button

Actions:

- Clicking the Housekeeping button will manually start the housekeeping process SystemManagement.general.housekeeping. Normally the housekeeping process runs automatically at certain intervals (TBD).
- Unless the underlying software has its own mechanisms, the housekeeping process will take care of:
  - sending a batch of e-mail notifications,
  - logging out users who were not active during a certain interval (see system parameter SYSTEM.LOGOUT) by calling the SystemManagement.User.Logout function.
- Clicking the Close button returns the user to the System Administrator page.

5 Functions

This previous chapter (Site Map) contains the major part of the functional description of the program logic. In some cases it refers to functions. These functions are described in this chapter.

The Functions Chapter should be considered as a specification of functionality, not as literal requirements on program function names and boundaries.

5.1 Learning Content Management System (LCMS)

The sub-system for managing content, the equivalent of a Learning Content Management System, will be split-up into three sub-sub-systems as well:

- playground content management system - an interactive environment that enables playground authors and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) playground objects.
- game content management system - an interactive environment that enables game authors and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) games.
- game manifest content management system - an interactive environment that enables game manifest composers and librarians to store, retrieve, copy, rename, delete, share, unshare (future: import and export) game manifests.

5.1.1 Game

Copy

Purpose:

To copy a game model.

Input:

A given game.

Process:

- The source item is copied and remains unchanged.
- The status of the new game is set to 1 = under construction.
- The name of the new game is set to "Copy of" (dialogue language dependent) plus the name of the source item.
- The current user becomes the owner of the new item.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new game.

Save

Purpose:

To save a game model.

Input:

- a given game model
- a temporary game model

Process:

- The temporary game model is checked for the following conditions:
  - name of the game has been filled in (error)
  - description has been filled in (error)
  - activities refer to existing roles only (error)

    - all activities are used in gameplay (warning)
    - all roles have at least one activity in gameplay (warning)
- If an error condition occurs and the status = 2 (private) or 3 (public), a message is sent to the user and the saving of the game is cancelled. In the case of status = 1 (under construction) or status = 4 (obsolete), the game is saved (see below).
- If a warning condition occurs, a message is sent to the user and the game is saved (see below).
- Saving in this case implies that the game model will be overwritten by the temporary model.

Output:

- OK in the case the game has been saved, NOT OK if not.

Edit

Purpose:

To create a temporary copy of a game model.

Input:

A given game.

Process:

The game is copied to a temporary datastructure for editing (parts of the) game and saving (or not) any intermediate changes.

Output:

A temporary game model.

5.1.2 Game manifest

Copy

Purpose:

To copy a game manifest.

Input:

A given game manifest.

Process:

- The source item is copied and remains unchanged.
- The status of the new game manifest is set to 1 = under construction.
- The name of the new game manifest is set to "Copy of" (dialogue language dependent) plus the name of the source item.
- The current user becomes the owner of the new item.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new game manifest.

5.1.3 Playground object

Copy

Purpose:

To copy a playground object.

Input:

A given playground object.

Process:

- The source item is copied and remains unchanged.
- The status of the new playground object is set to 1 = under construction.
- The name of the new playground object game is set to "Copy of" (dialogue language dependent) plus the name of the source item.
- The current user becomes the owner of the new item.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new playground object.

5.2 Learning Management System (LMS)

The sub-system for managing the learning, the equivalent of a Learning Management System, is an interactive environment that enables game masters to register/enrol participants, and to start, administer and stop game sessions based on a game manifest. In future versions it enables them to manage session logs + data, to arcive them (as a report or as a file) and to remove them from the system.

5.2.1 Session

Copy

Purpose:

To copy a game session.

Input:

- A given game session.

Process:

- The source item is copied and remains unchanged.
- The list of participants and their role allocation is copied as well.
- The name of the new session gets "Copy of" as prefix (language dependent).
- The status of the new session is set to 1 = session created, independent of the status of the source session.
- The owner of the new session is set to the current user.
- The last-mutated-by field will refer to the curremt user; the last- mutated-on field is updated.

Output:

- A new game session.

5.3 Virtual Learning Environment (VLE)

The sub-system for learning, the equivalent of a Virtual Learning Environment, enables game (session) participants (students, tutors) to participate to a game session, i.e. play a certain role. It further enables game masters to monitor sessions, even after they have been completed. Game masters can intervene during sessions by changing the role-allocation of participants and by taking over a role for some time.

5.3.1 Session

Status

Input:

Session-id, role-id

Process:

The system looks up the session data for a certain role.

Output:

The system returns:

- session name
- status
    1. 3 = Running (session is running)
    2. 4 = Finished (session is finished in normal way)
    3. 5 = Aborted (session is finished in abnormal way)
- role + character of the participant
- session start (date+time)
- session stop (date+time)
- game master (full name)

Step-of-play

Input:

Session-ID

Process:

Output:

visible name of the current step. If this is empty it is the sequence number of the step.

5.4 System management

Managing the system (not shown in figure) is an interactive environment that enables all system users to inspect and/or change personal settings, like dialogue language and notification preferences. It enables system administrators to register users and their privileges, to maintain a variety of dialogue languages, to maintain the interactive city map, and to manage all data involved. Automatically it does some housekeeping like sending out e-mail notifications.

5.4.1 General

Housekeeping

Purpose:

To execute housekeeping activities.

Process:

- When the system starts-up this function is executed. Normally it will then run at a certain interval (TBD) automatically. It also can be started manually.
- At system start-up and when run manually add an entry to the log by SystemAdmin.Log.Add.
- Always update the system variable for last date+time of housekeeping.
- Once in 24 hours:
    - Check directory for temporary files and remove files older than 24 hours.

5.4.2 Group

Copy

Purpose:

To copy a user group.

Input:

A given user group.

Process:

- The source item is copied and remains unchanged.
- All the members of the group are copied to the new group as well.
- The name of the new group gets a "Copy of" prefix (language dependent).
- The owner of the new group is set to the current user.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new user group.

5.4.3 LanguagePack

Copy

Purpose:

To copy a language item.

Input:

A given language item.

Process:

- The source item is copied and remains unchanged.
- All the members of the group are copied to the new item as well.
- The name (message key) of the new item is set to "Copy of" (language dependent) plus the name of the source item.
- The owner of the new group is set to the current user.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new language item.

5.4.4 Log

Add

Purpose:

To add an entry to the system log.

Input:

- name of the function that calls this function
- user-id
- optional parameters, e.g. in the case of a game session:
  - session name
  - role
  - action (send message, upload file, execute activity (send message, upload file, progress)

Process:

The system adds a record to the log table. It contains:

- name of the function that called the log function,
- user-id,
- date+time,
- optional parameters

Output:

- return value is OK (log record created) or NOT OK.

5.4.5 Parameter

Copy

Purpose:

To copy a system parameter.

Input:

- A given system parameter.

Process:

- The source item is copied and remains unchanged.
- The name of the new item gets a "Copy of" (language dependent) prefix.
- The owner of the new item is set to the current user.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

- A new system parameter.

5.4.6 User

Access

Purpose:

To check a user's access rights.

Input

Role-id

Process:

The system checks whether the Role-id is part of the login- session.settings. If so it returns OK, else NOT OK.

Output:

return value is OK or NOT OK.

Changepwd

Purpose:

To process a change password request.

Input:

user-id, new password

Process:

The system checks whether the new password conforms any rules (minimum length). If so,

- the password is encrypted and saved in the user settings
- an e-mail notification is planned
- the action is registered in the login-session log,
- the function returns OK,

If not, the function returns NOT OK.

Output:

return value is OK or NOT OK.

Copy

Purpose:

To copy a user.

Input:

A given user.

Process:

- The source item is copied and remains unchanged.
- The login name of the new item gets a "Copy of" prefix (language dependent, converted to lower case).
- The owner of the new item is set to the current user.
- The last-mutated-by field will refer to the current user; the last- mutated-on field is updated.

Output:

A new user.

Login

Purpose:

To process a login request.

Input:

- login name
- password

Process:

The system checks the login name and the encrypted password against the user settings table.

If login name plus password have a match and the user has status active:

- the system writes actual date+time to the last-login field in the user settings table
- the system logs the event by calling SystemManagement.Log.Add
- the system creates a login-session, determines the user's privileges (access rights) and stores them for easy reference as login- session.privileges
- the system returns OK

else

- the system returns NOT OK.

Output:

- return value is OK or NOT OK

Logout

Purpose:

To process a logout request.

Input:

login-session id

Process:

The system checks whether the login-session exists. If so:

- the system logs the event by calling SystemManagement.Log.Add
- the system deletes the login-session
- the system returns OK

else

- the system returns NOT OK.

Output:

return value is OK or NOT OK

Notification

Purpose:

To send an e-mail notification to a user.

Input:

1. User-id
2. message key
3. parameters 1-7

Process:

- The system composes a message based on the message key. It substitutes %1% - %7% placeholders by the values of parameters 1-7. The message is send to the e-mail address belonging to the user-id.

Output:

- E-mail message sent.

Upload

Purpose:

To register a batch of users.

Input:

A CSV file with:

- first record: a list of field names
- each following record: data for one of the new users
- the field names are:
    - given name (required)
    - family name (required)
    - login name (required)
    - Game session master (Y/N, default: N)
    - Game manifest composer (Y/N, default: N)
    - Game Author (Y/N, default: N)
    - Playground author (Y/N, default: N)
    - LCMS administrator (Y/N, default: N)

        - LMS administrator (Y/N, default: N)
        - VLE administrator (Y/N, default: N)
        - groups (a list of group names the user belongs to separated by "|", may be empty)
        - dialogue language, default "nl"
        - e-mail address (required)

Process:

- The system processes the input file line-by-line while producing an output file with status and error messages:
    - the input line is copied to the output file
    - if required fields in the input line are missing, an error message plus the name of the first missing field is written to the output file and further processing of the input line is skipped
    - if the login name (automatically converted to lower case) exists already, an error message is written to the output file and further processing of the input line is skipped
    - if one of the group names does not exist, an error message plus the name of the first non existing group name is written to the output file and further processing of the input line is skipped
    - if the e-mail address (automatically converted to lower case) exists already, a warning message is written to the output file
    - if non required fields are missing, they will get their default value
    - the password will be system generated
    - the status will be set to Enabled
    - the data are added to the User table
    - the user will receive an e-mail notification via SystemManagement.User.Notification (with the system URL, login name and the password as parameters).
- The output file will be closed and written to a temporary directory.
- The whole operation results in a single entry in the system log via SystemManagement.Log.Add.

- Output:
    - A link to a text file.

6 User interface

The User Interface chapter describes shortly the interface guidelines for Cyberdam 2.0.

6.1 Navigation

- The web pages are hierarchically ordered. To avoid confusing the users, cross- linking (linking outside the page hierarchy hierarchy) will be kept to the minimum.
- Each page contains breadcrumbs for easy navigation upwards.
- Each page contains a help button, that opens a popup window with language dependent and context sensitive help.
- After login, each page contains the full name of the user (clickable). When clicked, the User settings page is displayed.
- Pages only will be visible for users who have access rights to them.
- Although a web interface is used, interaction follows as much as possible the guidelines for windows user interface design.

6.2 Lay-out

- All formatting is done through stylesheets. In Cyberdam 2.0 stylesheets are defined at system level only, i.e. users cannot implement their own stylesheets.
- All Cyberdam 2.0 pages have the same style. The only exceptions are the home pages of the Cyberdam objects. It is functional that they have their own styles.
- The title bar of the browser window displays the name of the server, i.e."Cyberdam 2.0.* Production Server" plus the actual page name.
- The style is sober, professional, neutral colours.
- Page lay-out is optimised for regular windows (width=800 pixels) on PC- displays.
- Page height is not fixed, although more than one long list per page will be avoided; scroll bars are not used as they may cause printing problems.

6.3 Tables

Cyberdam 2.0 frequently uses tables. They should all have the same basic functions:

- Selection of columns and their sequence is fixed.
- Tables may be sorted on each of their columns (ascending and descending); exception: columns with calculated fields.
- A table without rows displays a meaningful explanation (e.g. "No game sessions are available yet").
- The user parameter Records-per-page determines how many records are displayed at a time.

6.4 Dialogue Language

The texts that the system displays are either belonging to the content of a certain simulation model or generated by the Cyberdam 2.0 system itself. The language of the former texts depends on the language used by their creators. The language of the system generated texts depends on user preferences. At least, English and Dutch texts are available for the Cyberdam 2.0 functions.

6.5 Editing

When a user may enter a description (long text), he will always use a WYSIWYG component that produces (limited) XHTML.

7 Data specification

A detailed data specification lies beyond the scope of this document. This chapter therefore only gives some general principles and specific comments intended to clarify the picture of the whole.

7.1 Data model

A game session refers to:

- a game manifest, which refers to
    - a game, which refers to:
        - roles
        - activities, of which there are three types:
            - message type
            - file upload type
            - progress type
        - steps-of-play
        - and connects activities to roles and steps-of-play via the gameplay
    - a playground with playground objects
    - and connects each role to a playground object
- a list of participants and their roles
- a session log and other during the session produced data (files)

7.2 Tables

7.2.1 Generic fields

Structure:

Most tables have six generic fields maintained by the system:

- ID - system provided unique key, however to facilitate import and export (in future versions of Cyberdam) each record needs a unique key based on data field(s) as well.
- created by - the user-id of the user who created the record
- created date+time - date+time of record creation
- last mutated by - the user-id of the user who most recently changed the record
- last mutated date+time - date+time of the last record change
- owner - the user-id of the user who is considered to be the owner of the record

Notes:

- When copying a record:
    - the original record is not changed
    - the new record has the last-mutated-on and last- mutated-by fields updated
    - the user who copied the record becomes the owner of the new record
    - usually more rules apply, e.g. copying a game model will change the status of the copy; these will be explained in the appropriate table section.

7.2.2 Playground object

Structure:

- URI, e.g. "sd- 012", must be unique
- object name
- object category
- object caption (max. 127 chars),
- object thumbnail
- object picture
- object map coordinates (x, y)
- visible on map? (Y/N)
- listed in directory (Y/N)
- status
- (generic fields)

Notes:

- Status:
    - 1 = under construction
    - 3 = public (available for map and/or directory)
    - 4 = obsolete

- Object category
    - 0 = TBD
    - 1 = government body
    - 2 = utility
    - 3 = enterprise
    - 4 = association
    - 5 = private household
    - 6 = photo shot

7.2.3 Game

Structure:

- TBD, see Site Map chapter

Notes:

- Game status:

  1 = under construction

  2 = private (can only be copied or used by owner)

  3 = public (may be copied or used by other people)

  4 = obsolete

- Game activity status:

  1 = under construction

  2 = (not used)

  3 = enabled

  4 = obsolete

7.2.4 Game manifest

Structure:

- TBD, see Site Map chapter.

Notes:

- Status:

  1 = under construction

  2 = private (can only be copied or used by owner)

  3 = public (may be copied or used by other people)

  4 = obsolete

7.2.5 Game session(s)

Structure:

- TBD, see Site Map chapter.

Notes:

Game session status:

- 1 = In preparation (session has a name, a manifest and a game master)
- 2 = Ready-to-start (session has participants allocated to roles)
- 3 = Running (session is running)
- 4 = Finished (session is finished in normal way)
- 5 = Aborted (session is finished in abnormal way)
- 6 = Cancelled (session is cancelled before it was running)

7.2.6 Session log + data

Structure:

- name of the function that calls this function
- user-id
- optional parameters, e.g. in the case of a game session:
    1. session name
    2. role
    3. action (send message, upload file, execute activity (send message, upload file, progress)
- date+time

- Notes:
    - -

7.2.7 User settings

Structure:

The user settings have been subdivided into five categories: basics, privileges, groups, preferences and statistics:

- basics (visible for the user, changeable by User administrator and System administrator)
    - given name
    - family name
    - login name
    - password (never visible)
    - status (current, deleted)
- privileges (visible for user, changeable by System administrator)
    - Game session master (Y/N)
    - Game manifest composer (Y/N)
    - Game Author (Y/N)
    - Playground author (Y/N)
    - LCMS administrator (Y/N)
    - LMS administrator (Y/N)
    - VLE administrator (Y/N)

    - User administrator (Y/N)
    - System Administrator (Y/N)
- groups (visible for user, changeable by User administrator and System administrator)
    - ...
- preferences (changeable by user)
    1. dialogue language
    2. e-mail address (used for sending notifications)
- statistics (maintained by the system)
    1. last login (date+time)

7.2.8 User groups

Structure:

- group category
- group name
- description
- # of members (calculated field, readonly)
- (generic fields)

Notes:

- group category is provided for ordering purposes only
- group name has to be unique

7.2.9 System parameters

Structure:

- parameter category (short text)
- parameter name (short text)
- default value (short text)
- actual value (short text)
- message key (short text)
- generic fields

Notes:

- The full name of a parameter is category.name. It has to be unique.
- The message key (optional) may refer to some comment text in the language pack.

7.2.10 Language pack

Structure:

- category (short text)
- message key (short text)
- text (short text or long text)
- generic fields

Notes:

- The category is for ordering the list of language items only.
- The message key has to be unique.
- The text either can be a single word or an XHTML text.
- In Cyberdam 2.0 two language packs are provided: Dutch (nl) and English (en).

7.2.11 Log

Structure:

- module, name of the calling function
- action
- parameter
- created by (user-id)
- created on (date+time)
- (no further generic fields)

- Notes:
  - -

7.3 System variables

The system maintains a set of variables:

- system
    - version (hard wired in program code)
    - copyright (hard wired in program code)
    - start (date+time)
    - lasthousekeeping
- for each login-session
    - user-id
    - login time+date

7.4 System parameters

The system parameters defined so far are:

- LCMS.MAPOBJECTS, a list of links between playground objects and city map locations.
- SYSTEM.CRON, initial value 10 minutes, the time interval between two subsequent housekeeping jobs.
- SYSTEM.EMAIL, initial value system@cyberdam.nl, the e-mail address used as sender of notifications.
- SYSTEM.LANGUAGE, initial value "en", the default dialogue langue.
- SYSTEM.LOGOUT, initial value 1 hour, the time interval of no-activity resulting in a system initiated logout.
- SYSTEM.ROWS, initial value = 20, the number of rows per page when displaying a table.
- SYSTEM.STYLESHEET.*, the various stylesheets involved (eventually another mechanism to store stylesheets maybe implemented).
- VLE.ACTIVITIES, initial value 5, the maximum number of activities in the new activities container on the session home page.
- VLE.FILES, initial value 5, the maximum number of files in the new file uploads container on the session home page.
- VLE.MAIL, initial value 5, the maximum number of messages in the new messages container on the session home page.
- VLE.MESSAGE, initial value Yes, the default setting for e-mail notifications on receiving new messages.
- VLE.STEP, initial value YES, the default setting for e-mail notifications on moving to an new step-of-play.