PS Software Engineering I
WS 2001/02
December 19, 2001

InfoWarrior
Version 0.1
Individual Document

Albert Meixner (albert@subnet.at)

Academic Supervisor: Dr. Schwaiger Roland

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Abstract

This document contains analysis and design of a computer based collecting and presenting information system with the working title "InfoWarrior".

Contents

Chapter 1

System Requirements, CRC Session and Essential UI

1.1 Introduction

The CRC Session was done to structure the project into relevant parts, which can be viewed independently. The resulting classes are a first step to an implementation and allow easier discussion on the required functionality of the system.

1.2 System requirements

- InfoWarrior is a system for creating, maintaining and distributing news electronically. It is both used by the news agency staff and the consumers. To allow this the system will provide different interfaces and support multiple media types.
- News are created by reporters and then released by editors. There are around 5 editors, each working with up to 50 reporters. A reporter writes 1 to 5 articles a day. Thus, the system must handle at least a load of 1250 new articles a day.
- Articles are textual, but can contain different media attachment, such as pictures, movies and sounds.
- Recent articles have to be accessbile immediatly, while older news may be stored in a slower archive.
- Articles are organized in semantical categories, such as Politics, Sports and Economy.
- Customers can access the system in multiple different ways
  - **Pull** includes services such as the WWW and WAP
  - **Push** allows notification through eMail, SMS, Instant Messaging and other message services.
- Not all information is available to all customers
  - There is a pool of basic information which is free for everyone

  - Relevant and recent information is organized in different prizing levels. It can only be viewed by customers, that subscribed to the matching prizing level within the category of the article.

- Both subscription to higher pay level and push services requires a user registration. Registered users will also be able to customize the interface to match their interests.

- New information can not only be entered by the staff, but also be imported from other external news systems. InfoWarrior has to provide an interface to allow import from such systems.

1.3 CRC Card Session

1.3.1 General

The actual CRC classes for the project are listed in the next chapter. In this section a list of objects and actions are given, that were used to find classes and methods in the system. Additionally we developed scenarios, which can be used to see if the selected classes are sufficient to represent the system.

1.3.2 Objects

- application
- user interface
- article
- category
- reporter
- editor
- administrator
- user
- customer
- prizing level
- external system
- storage
- archive
- subscription

1.3.3 Actions

- create news
- maintain news
- import news
- pull news
- push news
- subscribe
- import news
- export news
- register user
- archive news

1.3.4 Scenarios

- Customer views information via the web
- Customer registers to the service and subscribes for payment levels
- Reporter creates and edits articles
- Editor views, edits and releases articles
- News are imported from an external system

1.3.5 Essential Use Cases

These use casses describe the essential actions that are execute in the above scenarios.

- Customer visits the InfoWarrior start screen
- Customer views news in a category
- Customer searches for a specific article
- Customer views a single article
- Customer registers
- Registered customer logs into the service
- Registered customer subscribes to a new payment level
- Registered customer subscribes to a push service

- Reporter creates a new article
- Reporter edits an existing article
- Editor looks for submitted articles
- Editor releases an article
- Editor edits an article
- Editor deletes an article

Chapter 2

CRC Classes

2.1 Classes

The resulting CRC Classes.

User		
Responsibilities Represents human user	**Collaborators** User Interface	**Subclasses** Customer Editor Reporter Administrator

Reporter		
Responsibilities Write Articles	**Collaborators** Article User Interface	**Superclasses** User

Editor		
Responsibilities Release Articles	**Collaborators** Article User Interface	**Superclasses** User

Customer		
Responsibilities	**Collaborators**	**Superclasses**
Read Articles	Article	User
Register	User Interface	
Subscribe		

Administrator		
Responsibilities	**Collaborators**	**Superclasses**
Manage Users	Article	User
Manage Categories	User Interface	

External System		
Responsibilities	**Collaborators**	**Subclasses**
Provide News	External Systems Interface	User Interface

External Systems Interface		
Responsibilities	**Collaborators**	**Subclasses**
Handle External Requests	External System	
Convert Input to Articles	Article	
Format Articles for Output	Core	
	Request	
	Response	

User Interface		
Responsibilities	**Collaborators**	**Superclasses**
Provide Graphical Interface for users	User	External System
	External Systems Interface	

Request		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Represent request to core	User Core	
	Message Core	
	Article	
	External Systems Interface	

Response		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Represent response from core	User Core	
	Message Core	
	Article	
	External Systems Interface	

Article		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Represent an Article in the system	Message Core Response Request External Systems Interface Category	

Category		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Structure stored Articles	Message Core Article Access Control	

Login		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Internal representation of a user	User Core Access Control	

Access Control		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Limit accesses to ressources	Login Category Article Message Core User Core	

User Core		
Responsibilities	**Collaborators**	**Sub/Superclasses**
Manage Logins	Login User Storage Access Control Message Core External Systems Interface Request Response	

Message Core		
Responsibilities Manage Articles	**Collaborators** Articles Message Storage Category Access Control User Core External Systems Interface Request Response	**Sub/Superclasses**

Message Storage		
Responsibilities Store Articles	**Collaborators** Article Category Message Core	**Subclasses** Message Archive

Message Archive		
Responsibilities Store old Articles	**Collaborators** Article Category Message Core	**Superclasses** Message Storage

User Storage		
Responsibilities Store User Information	**Collaborators** Login User Core	**Sub/Superclasses**

Chapter 3

Use Cases Description

3.1 Actors

- Editor
- Reporter
- Customer
- External System
- Administrator
- InfoWarrrior Application
- Administrator
- News
- InfoWarrior Search Machine
- Storage Unit

3.2 Scenarios

- Subscribe to InfoWarrior Services
- User logs in
- User searches News
- Unsubscribe from InfoWarrior Services
- Customer edits Preferences
- Administrator creates a User
- Administrator searches a User

- Administrator modifies a User
- Administrator creates a Category
- Administrator deletes a Category
- Editor edits News

3.2.1 Description

1. **Name:** Subscribe to InfoWarrior Services
   Description: Somebody wants to subscribe to InfoWarrior services. She/he opens the official InfoWarrior web-page and selects "Sing up now!" link and enters necessary information (username, password, e-mail address,...) in order to create an account. The user can decide weather she/he wants to use just free or billable InfoWarrior services as well. After submitting this information she/he becomes a user of the corresponding InfoWarrior services.

2. **Name:** Unsubscribe From InfoWarrior Services
   Description: A user wants to unsubscribe from InfoWarrior Services. She/he selects "Unsubscribe From InfoWarrior Services" link. Then she/he will be asked does she/he want to unsubscribe from free InfoWarrior services, billable InfoWarrior services or from the both (free and billable) services. After submitting her/his answer, she/he won't be a user of corresponding InfoWarrior services anymore.

3. **Name:** User logs in
   Description: A user (editor, reporter, Customer) wants to log in. She/he opens the official InfoWarrior web-page and enters her/his user name and password in the corresponding fields. If a user (editor, reporter, Customer) enters incorrect user name or password a login incorrect window will be opened, and a user can try to log in again.Upon successful login new User Interface is shown, with features available depending on access level. That means, if for example, a reporter logs in, a reporter specific window will be opened.

4. **Name:** Customer edits Preferences
   Description: An Customer wants to edit her/his preferences. After login every user has a link "Edit Preferences". When she/he selects this link, a page with her/his preferences will be opened. The preferences page contains user specific information : user name, encrypted password, e-mail address, mobile phone number, etc., that is, it contains all information given upon subscribing to InfoWarrior services. The user can now modify her/his preferences. After editing, she/he selects "Submit changes" in order to apply changes to her/his account.

5. **Name:** Editor edits News
   Description: She/he wants to add, delete, or alter news. Upon login every editor has buttons for each of these actions. When she/he selects one of these buttons, she/he can browse the files, that she/he wants to add/delete/alter. After editing she/he has to commit "Submit changes" in order to apply changes to the system storage unit.

6. **Name:** User searches News in the Primary Storage System
Description: A user(editor, reporter, Customer) wants to search for recent news. She/he selects "Search" link in order to start search page. A user enters wanted pattern in the corresponding field and selects "Start Search" button. The InfoWarrior Search Machine outputs all news corresponding to the searching pattern.

7. **Name:** User searches News in the Secondary Storage System
Description: A user(editor, reporter, Customer) wants to search for older news. She/he selects "Advanced Search" link in order to start the advanced search page. Afterwards she/he enters wanted pattern in the corresponding field and selects "Start Search" button. The InfoWarrior Search Machine outputs all news corresponding to the searching pattern.

8. **Name:** Administrator logs in
Description The administor opens the standard login page and provides the his loginname and password. After validating the login data, the administrator is shown the administrator main page, that provides links for essential administrative activities.

9. **Name:** Administrator creates a User
Description: The administrator wants to create a new reporter or editor in the system. From the admin main page he uses the create user page and enters the users essential data and privileges. After the admininistrator confirms the data, the system creates a new user.

10. **Name:** Administrator searches User
Description The administrator wants to view existing users. He enters search criteria for the users to be returned and selects search. The system returns a list of matching users, from which the administrator can select a user.

11. **Name:** Administrator modifies User Data
Description: The administrator wants to modify user data or privileges. He searches the specific user as described in "Administrator searches User". He selects to modify the user and the user modification page is returned. There he changes the data and confirms them by clicking a "submit button. The system commits the changes to the user database.

12. **Name:** Administrator deletes a User
Description: The administrator wants to delete a user. He searches the specific user as described in "Administrator searches User". He selects to delete the user. After confirmation from the administrator, the system user is removed from the user database.

13. **Name:** Administrator creates a category
Description: The administrator wants to create a new category for news. He selects the link on the main page and enters the name of the category. After the administrator confirms the data, the system creates a new category

14. **Name:** Administrator deletes a category
Description: The administrator wants to delete a category. He selects the link on

the main page and on the next page the category he wants to delete. He gets a warning how much news will be deleted on cascade if he will delete the category. After the administrator confirms the data, the system deletes the category and all news assigned to it.

Chapter 4

Classes

4.1 Defined Classes

- Storage
- UserStorage
- Message Storage
- Primary Message Storage
- Secondary Message Storage
- User
- Contact
- PostalContact
- Address
- PhoneContact
- EMailContact
- Customer
- Reporter
- Editor
- Administrator
- External System
- UserInterface
- Customer Interface
- ReporterInterface

- EditorInterface
- AdministratorInterface
- SystemInterface
- MediaObject
- Picture
- Video
- Audio
- Message
- Messageset
- Paylevel
- Category
- ACLSecuredObject
- SecuredObject
- ACL
- Permission
- Roles
- UserCore
- MessageCore
- Request
- Response
- XMLConverter

4.1.1 Storage

Abstract parent class for all storing purposes. It is an abstraction layer for different types of storage devices/server.
Methods:
Connect () : Boolean
Close () : Boolean
GetReport () : REPORT
GetSize () : Integer

4.1.2 UserStorage

User storage is a Storage class specialized for storing all user related information. This includes logins, subscriptions and security information.
Methods:
GetUser (id) : USER
AddUser (USER) : boolean
UpdateUser (USER) : boolean
DeleteUser (id) : boolean
GetRole () : ROLE
DeleteRole (ROLE) : boolean
UpdateRole (ROLE) : boolean
AddRole (ROLE) : boolean
AddPaylevel(PAYLEVEL): boolean
UpdatePaylevel (PAYLEVEL) : boolean
DeletePaylevel (id) : boolean

4.1.3 MessageStorage

Specialized storage class for messages related information, including messages, categories and media attached to articles
Methods:
GetMessage (id) : MESSAGE
AddMessage (MESSAGE) : boolean
FindMessage (REQUEST) : RESPONSE

4.1.4 PrimaryMessageStorage

The main storage for messages in the system. This storage has to be quickly accessible.
Methods:
UpdateMessage (MESSAGE) : boolean
DeleteMessage (id) : boolean AddCategory (CATEGORY) : boolean
UpadteCategory (CATEGORY) : boolean

4.1.5 SecondaryMessageStorage

Storage for older messages, serves as an Archive

4.1.6 User

Abstract class for storing real person users of the system. This information is used for payment and management purposes.
Public Attributes:

Name
Address
Contact
CreditCard
Password
Subscriptions
LastLogin
Birthdate
Company
Operating System
Browser
Homepage
Login
Contacts

4.1.7 Contact

Abstract base class for all contacts of a person. Subclasses describe different types of contacts. **Attributes:**
ValidityDate
Methods:
Print

4.1.8 PostalContact

Contact class for snail mail. **Attributes:**
Address

4.1.9 Address

A streetaddress as used by Postalcontact **Attributes:**
Street
Number
PostalCode
City
Region
Country

4.1.10 PhoneContact

Contact class for telephone services. **Attributes:**
PhoneNumber

4.1.11 MobilePhoneContact

Contact class for mobile telephone services.

4.1.12 FaxContact

Contact class for fax.

4.1.13 EMailContact

Contact class for electronic mail services. **Attributes:**
MailAddress

4.1.14 Customer

User subclass for Customers of the System.
Public Attributes:

4.1.15 Reporter

Subclass for Reporters.
Public Attributes:
Department
Superior
Methods:
getArticles(): MESSAGES
getArticlesPending: MESSAGES

4.1.16 Editor

Class for editors of the system.
Attributes Reporters

4.1.17 Administrator

Class for administrators of the system.

4.1.18 External System

Abstract class for all external systems. The implementations will vary a lot from system to system.

4.1.19 UserInterface

Base class for all UserInterfaces. A subclass of External System. Represent any kind of interactive, graphical interface.

4.1.20 CustomerInterface

The user interface shown to customers of the system.

4.1.21 ReporterInterface

The user interface used by reporters.

4.1.22 EditorInterface

The user interface used by editors.

4.1.23 AdministratorInterface

The user interface used by administrators.

4.1.24 SystemInterface

A common interface for all external systems to access the InfoWarrior functionality.
Methods:
executeRequest(REQUEST): RESPONSE

4.1.25 Media Object

Abstract class for storing media objects.
Public Attributes:
MIME Type

4.1.26 Picture

Subclass for Pictures.
Public Attributes:
Heigth
Width
Depth
Date

dpi

4.1.27 Video

Sublass for Videos.
Public Attributes:
Heigth
Width
Depth
Length
Date

4.1.28 Audio

Subclass for Audio files.
Public Attributes:
Length
SampleRate

4.1.29 Message

Class for storing articles and other types of news in the InfoWarrior system.
Public Attributes:
Author
Title
Abstract
Text
Date
Editor
Category
PayLevel
LastChanged
Released

4.1.30 MessageSet

A set of messages as returned by a search.
Methods:
GetMessages():MESSAGES

4.1.31 Paylevel

Describes different paylevels as used in the billing system.
Public Attributes:
Name
Cost

4.1.32 Category

Sematical category for structuring message storage
Public Attributes:
Name
Abstract

4.1.33 Secured Object

Interface for all objects, to which access is restricted in some way. Each class implementing this interface must know, who is allowed to execute operations on it.
Methods:
isAllowed (USER, ACTION) : boolean
GetParent()

4.1.34 ACLSecuredObject

An implementation of the SecuredObject interface using Access Control Lists.

4.1.35 ACL

A List describing pairs of roles and permissions
Methods:
GetPermission()
GetRoles()
HasPermission()

4.1.36 Permission

Permission for a certain operation.
Public Attributes:
Name

4.1.37 Roles

A role is a set of users, that have the same permissions in some context. This allows to grant permissions to ”functions”, rather than individual persons
Public Attributes:
Name
Methods:
GetMembers(): USER
AddMembers(USER): boolean
RemoveMembers(USER): boolean
isMember(USER): boolean

4.1.38 UserCore

The logic for user handling and security.
Methods:
executeRequest(REQUEST):RESPONSE

4.1.39 MessageCore

Logic for Message handling.
Methods:
executeRequest(REQUEST):RESPONSE

4.1.40 Request

Some kind of a request from an external system (e.g. FindMessage, GetMessage, ..)
Public Attributes:
Command

4.1.41 Response

A response created by the system, after receiving a request
Public Attributes:
Errorcode
Results

4.1.42 XMLConverter

Interface for converting Request/Reponses to XML and vice versa.
Methods:
XML2Request() Response2XML()

4.2 Class Diagrams

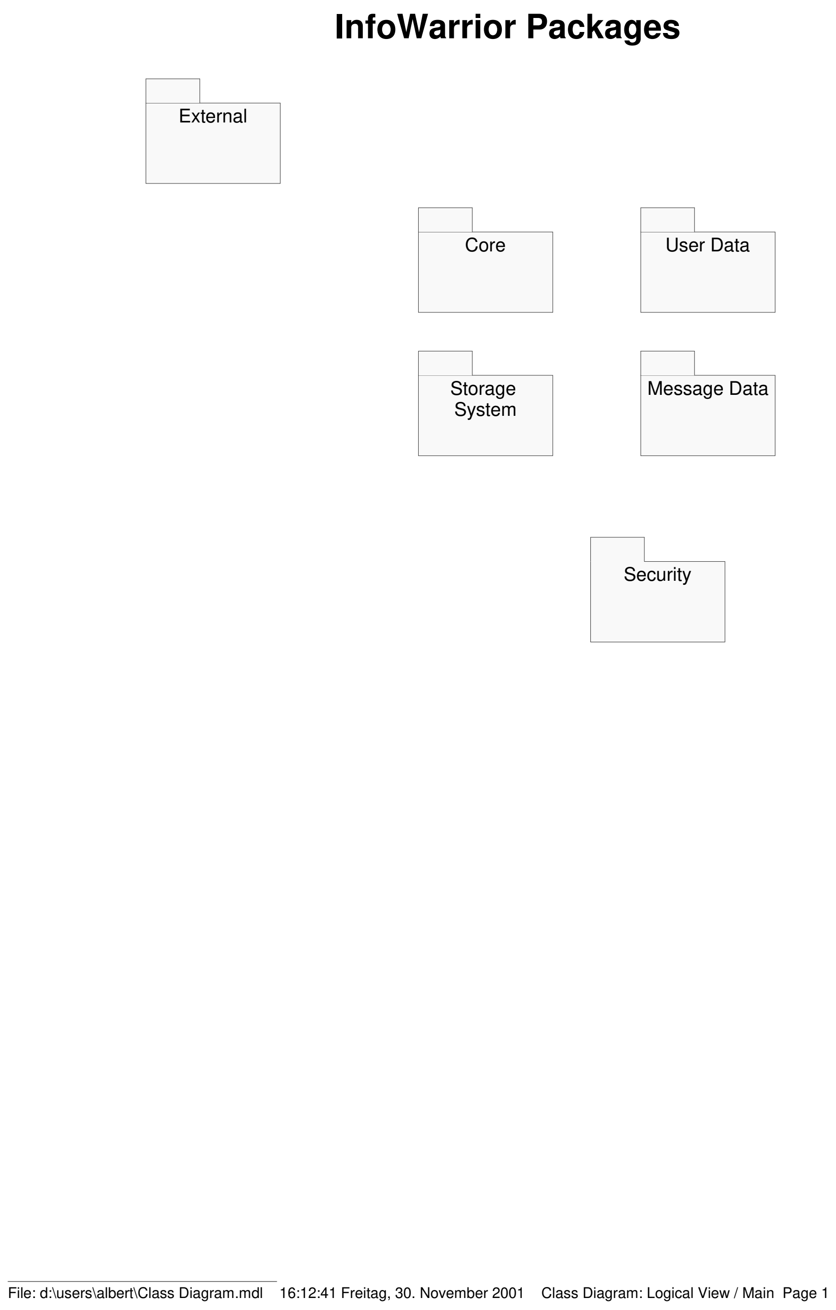
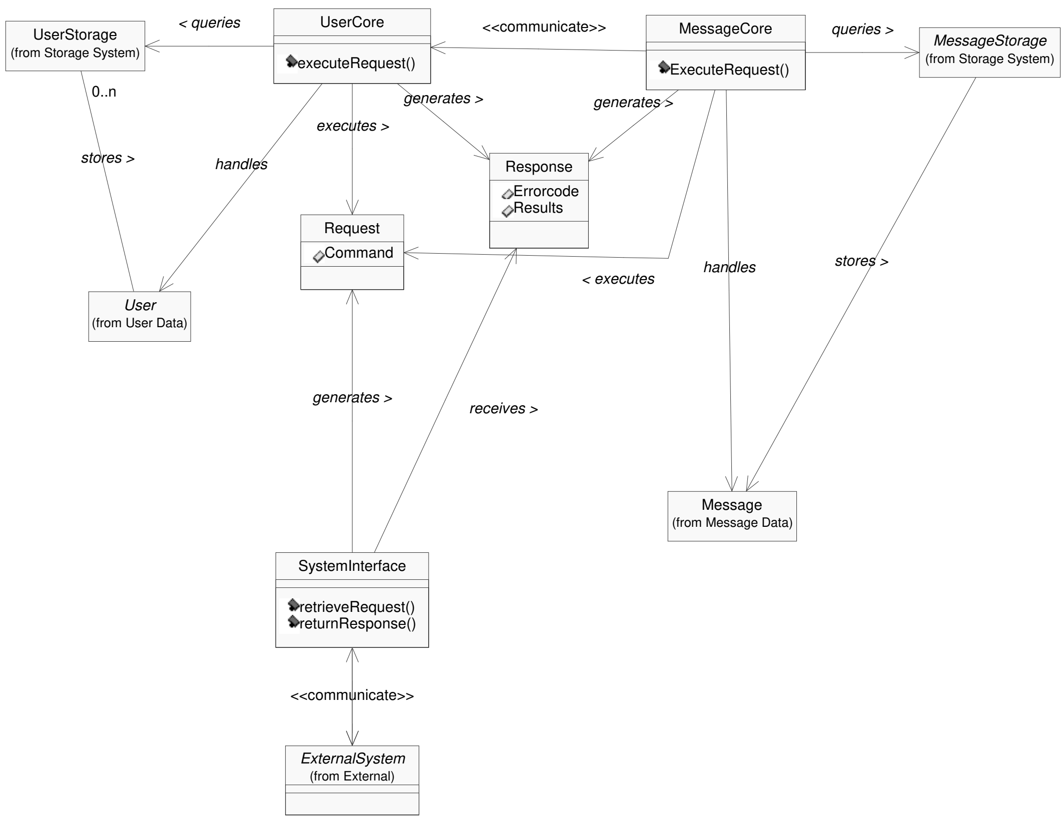


Table 4.1: All Packages

Core Package



File: d:\users\albert\Class Diagram.mdl 16:14:25 Freitag, 30. November 2001 Class Diagram: Core / Core Page 1

Table 4.2: Core Package

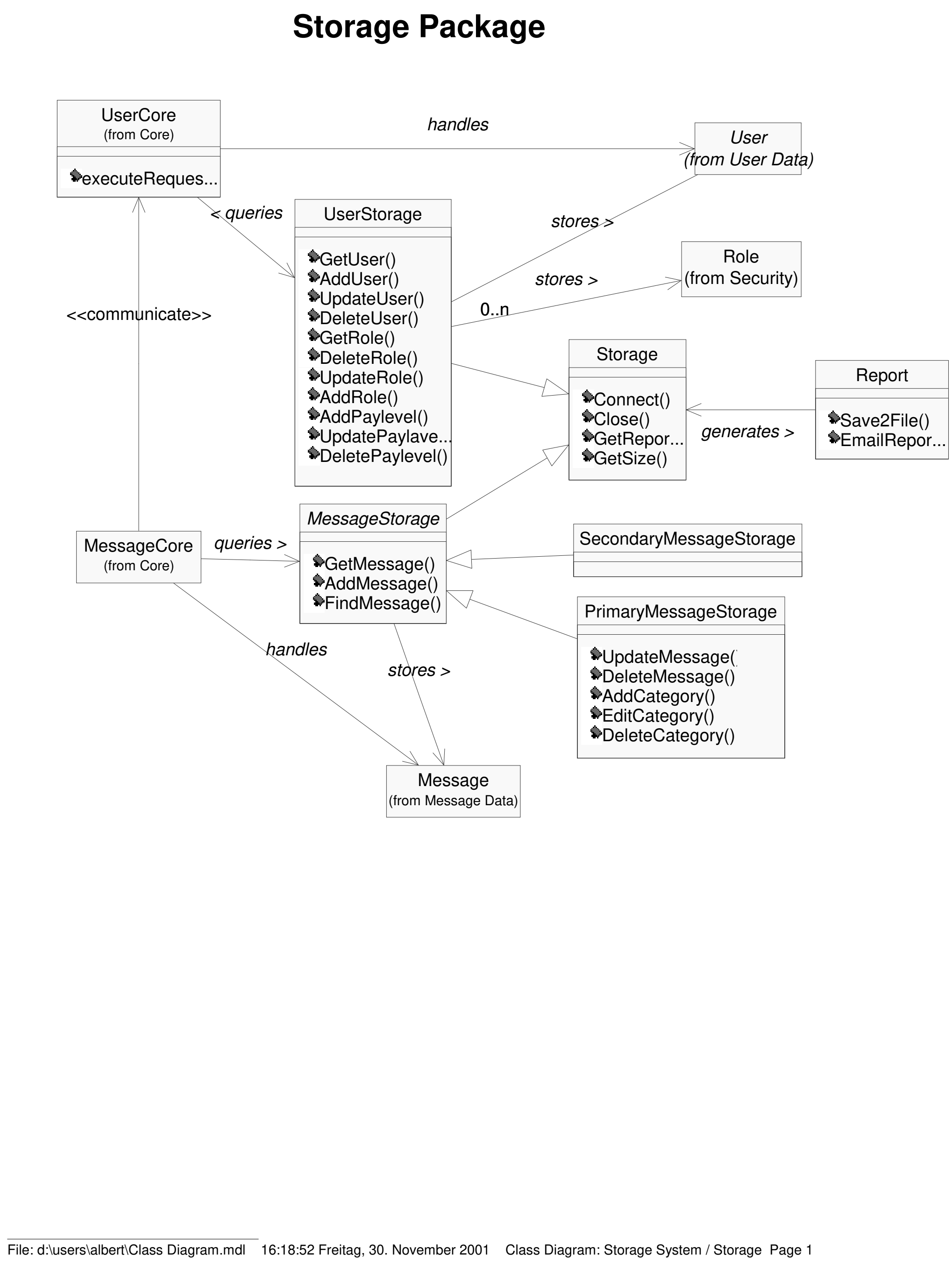


Table 4.3: Storage Package

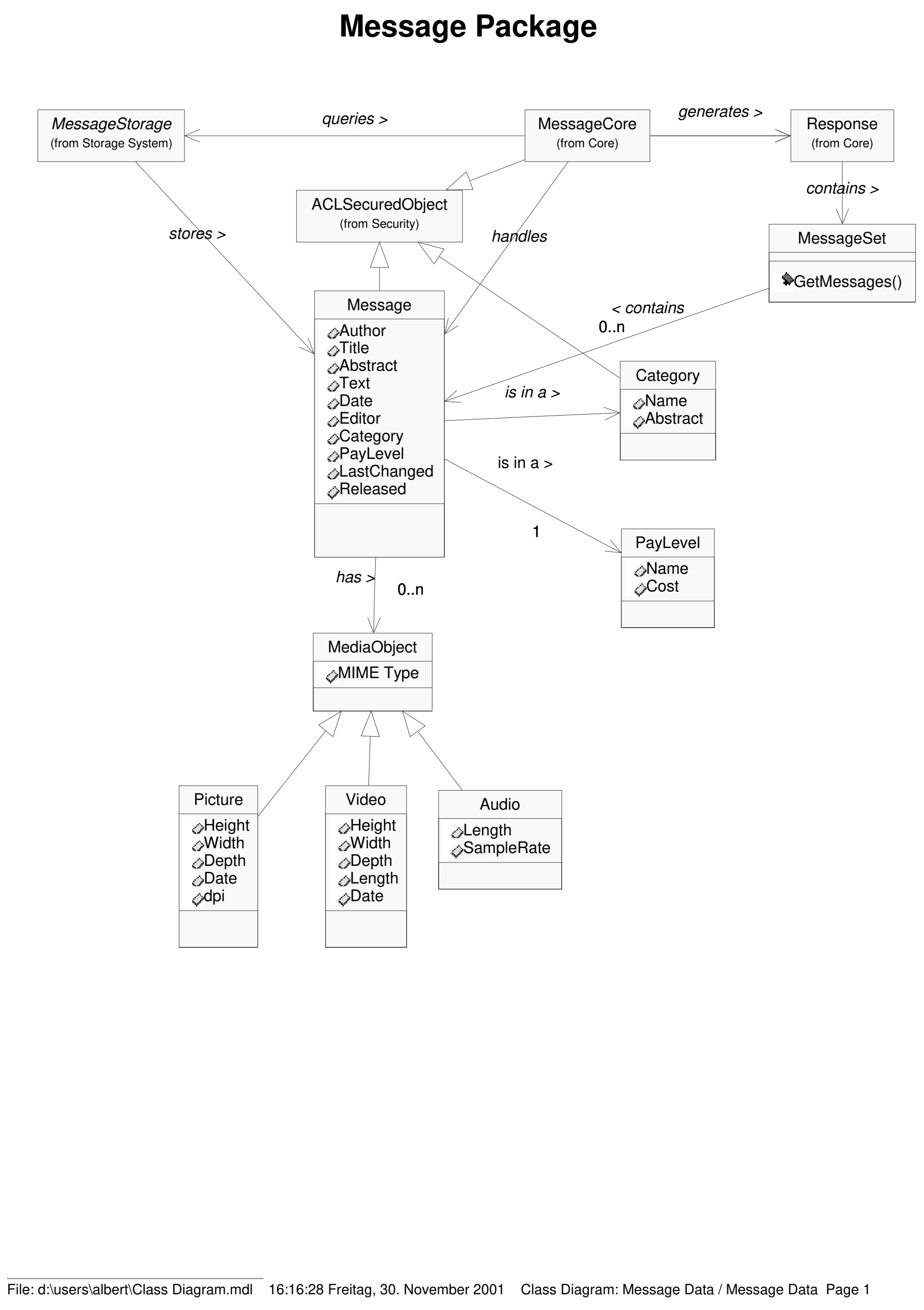


Table 4.4: Message Package

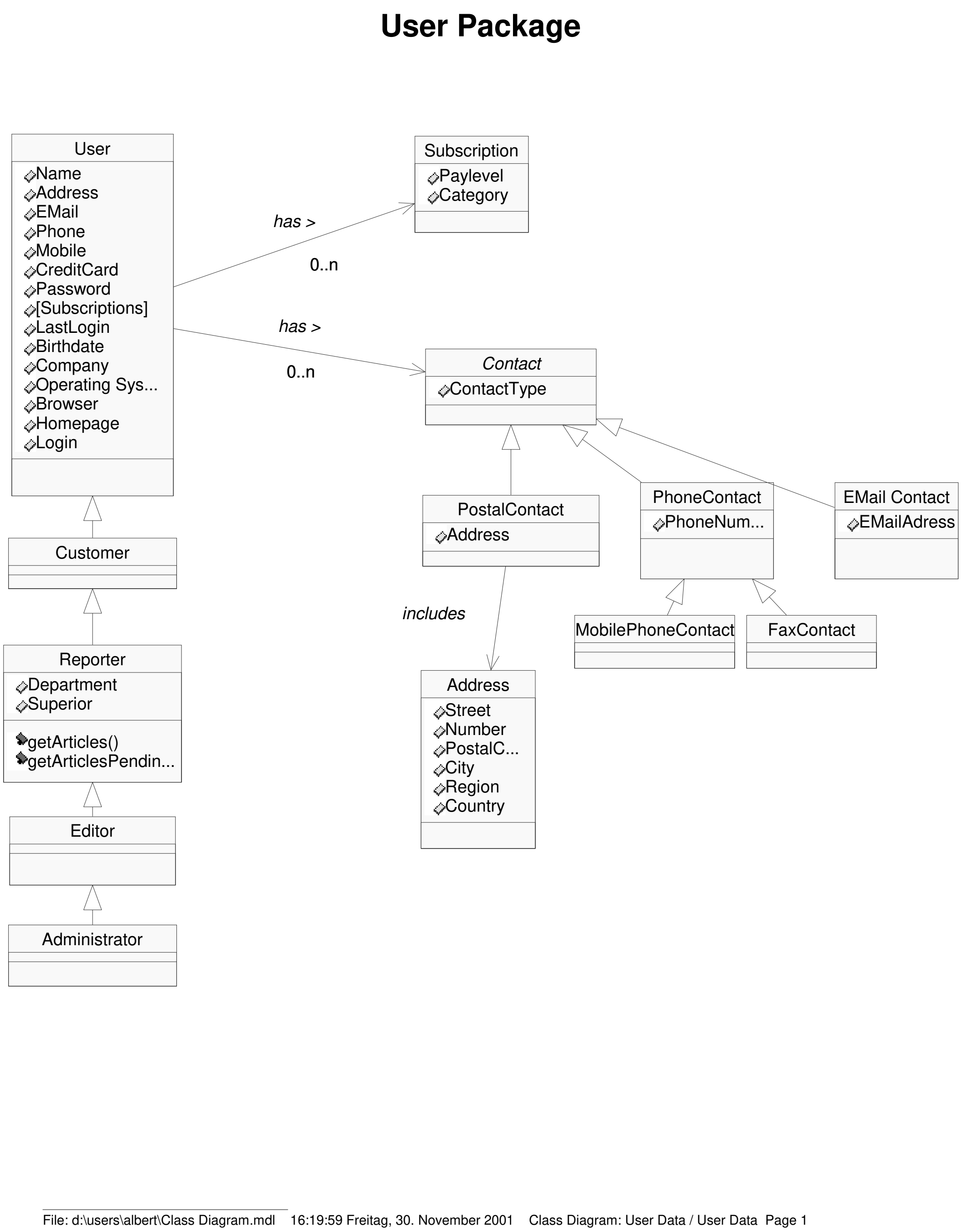


Table 4.5: User Package

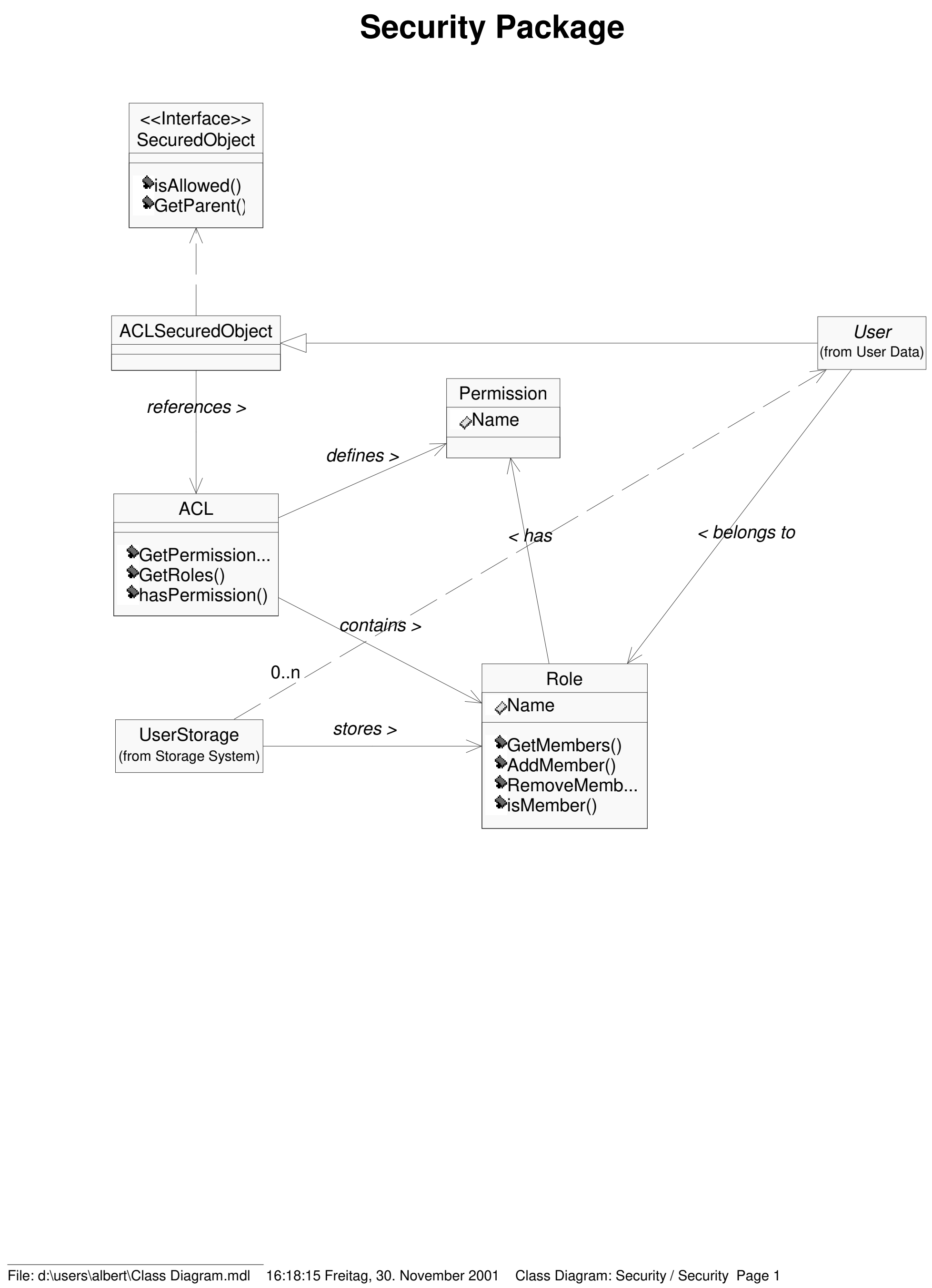
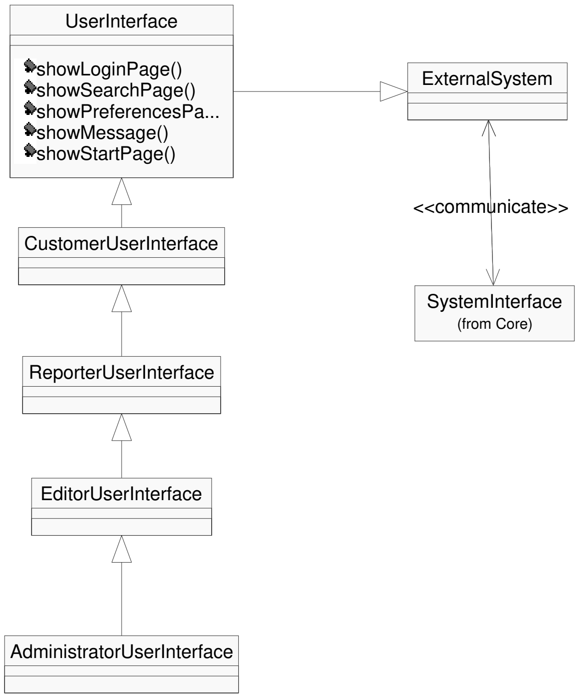


Table 4.6: Security Package

External Package



File: d:\users\albert\Class Diagram.mdl 16:15:34 Freitag, 30. November 2001 Class Diagram: External / External Page 1

Table 4.7: External Package

Chapter 5

Interface Prototype

5.1 HTML Prototype

5.1.1 Introduction

The HTML prototype is the major communication interface with the InfoWarrior system. Therefore it has to provide interfaces for all necessary tasks, including doucment editing and system administration. As this is exactly, what the essential ui design was done for, it could be modelled after the screens presented in the previous chapter.

As the functionality was already defined by the essential user interface, the focus of the prototype was to incorporate this functionality in a consistent and easy-to-use graphical interface.

5.1.2 Design Principles

- All pages should have a consistent look and feel.
- Important functions should be easily accessible, if possible with a single click.
- Each page must contain a navigation element, that allows the user to find out his current position in the site structure and to reach other important pages within the site
- For each section of the page (start page, category pages, ..) the most important content should be visible immediatly, lesser functions are reachable via a menu structure.
- The staff sites (editor, reporter, administrator) are designed to match the typical workflow. Important information and common tasks have to be visible on the main screen.

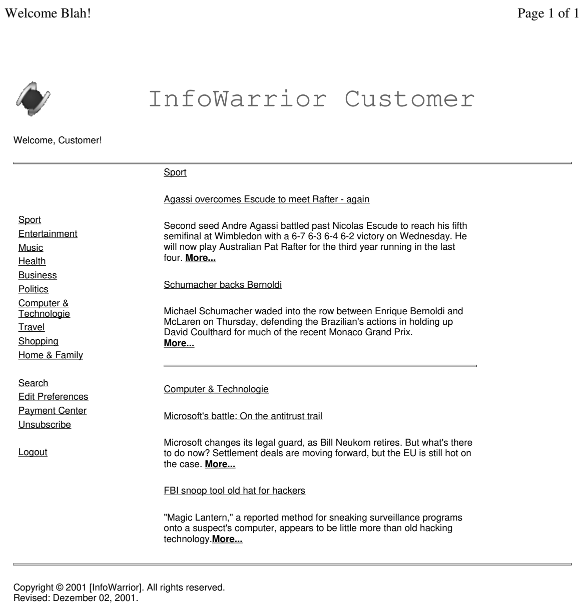


Figure 5.1: Customer Page

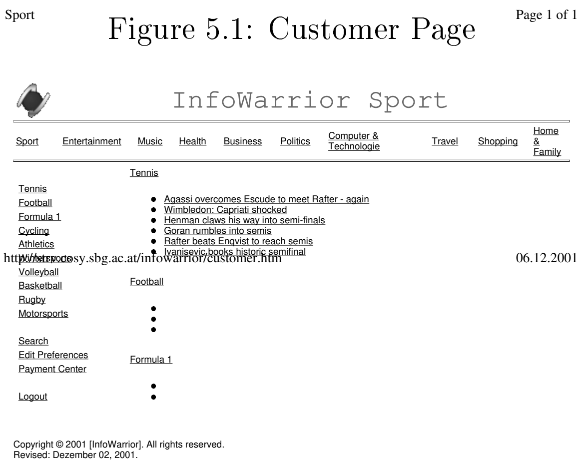


Figure 5.2: Sport Category

5.2 WML Prototype

5.2.1 Introduction

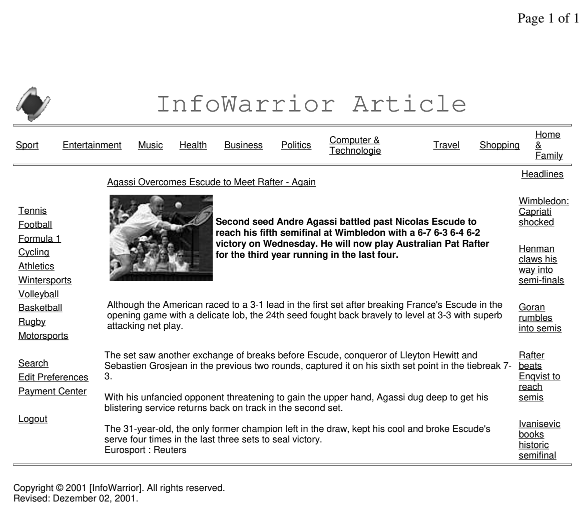
The WML prototype is baed on a reduced version of the essential user interface design. The original design had to be modified to fit on the small screen of a mobile device. Only the Client Interface was modelled in WML, as we agreed, that the staff interfaces would be unusable on a cell phone.

http://strv.cosy.sbg.ac.at/infowarrior/sport.htm 06.12.2001

5.2.2 Design Principles

We used a set of basic principles to get a usable desgin.

- The WML site should not be a port of the HTML page, but a completely seperate design
- The WML site should still be similar enough to the web site to allow users of the web page to utilize it without an extra learning process



Search Figure 5.3: Sport Article Page 1 of 1

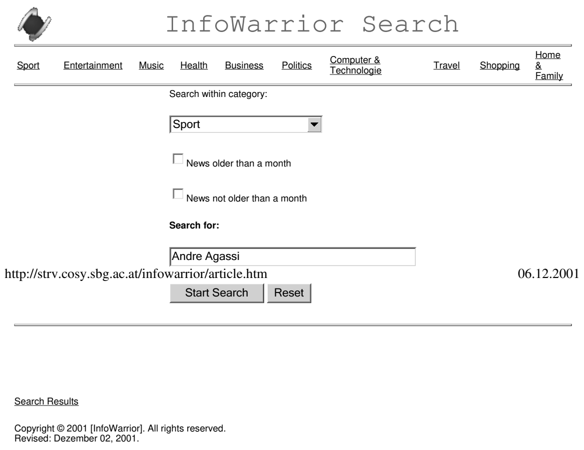


Figure 5.4: Basic Search

- All functionality of the system should be available via WAP
- All main items of the site should be reachable from every page
- Each page's information should be visible immediatly without scrolling. Navigation elements may never hide content
- Each page should have self-explanatory title and header

http://strv.cosy.sbg.ac.at/infowarrior/search.htm 06.12.2001

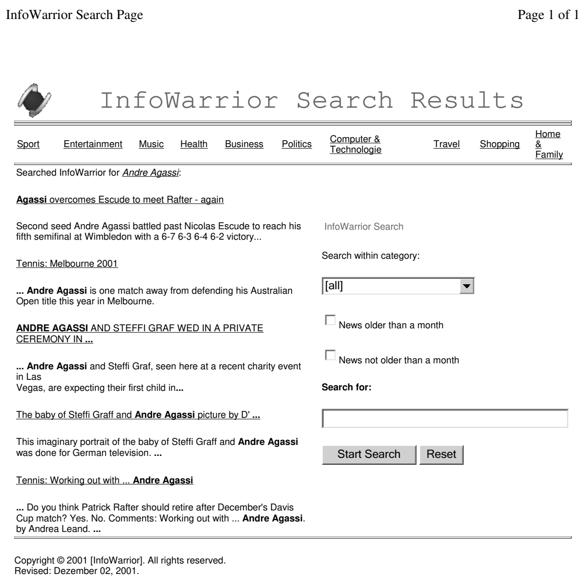


Figure 5.5: Search Results

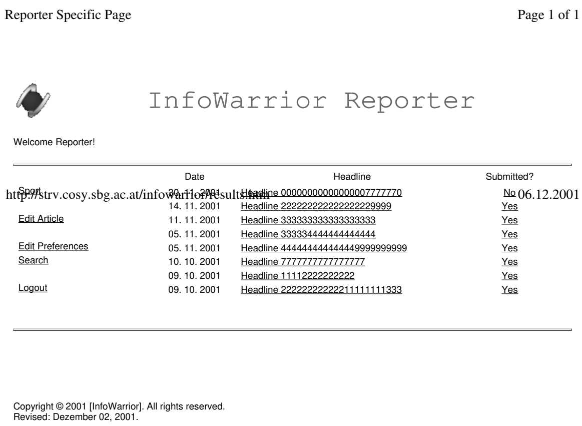


Figure 5.6: Reporter Interface

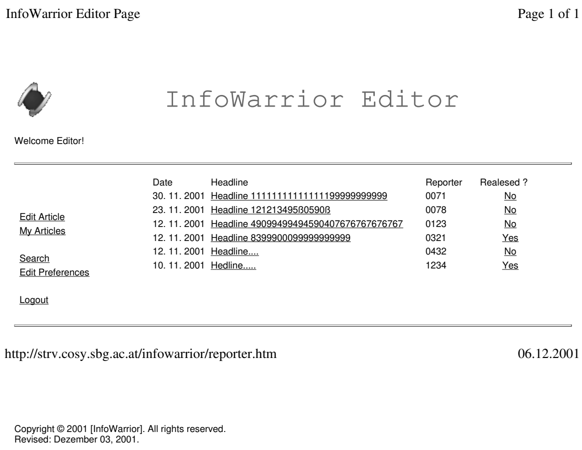


Figure 5.7: Editor Interface

InfoWarrior Admin Page　　　　Page 1 of 1

InfoWarrior Admin

Welcome Admin!

Create a User
Delete a User
Edit User Profile
Create Database
Edit Database
View System Log

Logout

http://strv.cosy.sbg.ac.at/infowarrior/admin.htm　　　　06.12.2001

Figure 5.8: Administrator Interface

Welcome

NewsWarrior
login
top news
categories
search
subscribe
about

Figure 5.9: Start Page

Welcome $user

NewsWarrior
Taliban claim leader Omar is safe
Indonesia arrests Suharto's fugitive son
U.S. envoys meet Arafat
----
categories
search
subscribe
about

Figure 5.10: Customer Page

Drugs

Drugs

Taliban claim leader Omar is safe
Indonesia arrests Suharto's fugitive son
8 more news in this category when you subscribe.

----
home
search
subscribe
about

Figure 5.11: Category

Drugs

Taliban claim leader Omar is safe

The Taliban on Wednesday denied that U.S. airstrikes had hit a leadership compound a day earlier and said the Taliban's supreme leader, Mullah Mohammad Omar, was safe and unharmed.

[full text]
----
other drug news
home
search
subscribe
about

Figure 5.12: Article

Search Results

Results for "Taliban" in

Taliban claim leader Omar is safe
[search again]

----
home
categories
subscribe
about

Figure 5.13: Search Results