PS Software Engineering I
WS 2001/02
December 4, 2001

InfoWarrior

Version 0.1

Task 5

Group Document

Team 4

Huskic Zenina (zehuskic@cosy.sbg.ac.at)
Theresa Krichhammer (Aphid@gmx.at)
Peter Gruber (pgruber@cosy.sbg.ac.at)
Albert Meixner (albert@subnet.at)
Michael Brachtl (mbrachtl@cosy.sbg.ac.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

1 Introduction **4**
1.1 Planning 4
1.2 Basic Organization 4
1.3 Goal 4
1.4 Responsibilities 5
1.5 Problem Statement 5
1.6 First considerations(CRC Session) 6
1.6.1 Identifying classes 6
1.6.2 Some scenarios 6
1.7 Open questions 7

2 CRC Session **8**
2.1 Problem Statement (Version two) 8
2.2 Open Questions 9
2.3 CRC Card Session 9
2.3.1 General 9
2.4 InfoWarrior CRC Session 10
2.5 Essential Use Case Modeling 11
2.5.1 General 11
2.5.2 Use Cases in InfoWarrior 12
2.6 Essential User Interface Prototyping (First considerations) 12

3 Use Cases Description **14**
3.1 Identifying Actors 14
3.2 Identifying Use Cases 14
3.2.1 Description 15
3.3 Essential User Interface Prototyping 17
3.4 Open Questions (3) 19

4 Classes **20**
4.1 Classes List 20
4.1.1 Storage 21
4.1.2 UserStorage 21
4.1.3 MessageStorage 22
4.1.4 PrimaryMessageStorage 22
4.1.5 SecondaryMessageStorage 22
4.1.6 User 22

4.1.7 Customer . . . 23
4.1.8 Reporter . . . 23
4.1.9 Editor . . . 23
4.1.10 Editor . . . 23
4.1.11 Administrator . . . 23
4.1.12 External System . . . 23
4.1.13 UserInterface . . . 23
4.1.14 CustomerInterface . . . 24
4.1.15 ReporterInterface . . . 24
4.1.16 EditorInterface . . . 24
4.1.17 AdministratorInterface . . . 24
4.1.18 SystemInterface . . . 24
4.1.19 Media Object . . . 24
4.1.20 Picture . . . 24
4.1.21 Video . . . 25
4.1.22 Audio . . . 25
4.1.23 Video . . . 25
4.1.24 Message . . . 25
4.1.25 MessageSet . . . 25
4.1.26 Paylevel . . . 26
4.1.27 Category . . . 26
4.1.28 ACLSecuredObject . . . 26
4.1.29 Secured Object . . . 26
4.1.30 ACL . . . 26
4.1.31 Permission . . . 26
4.1.32 Roles . . . 27
4.1.33 UserCore . . . 27
4.1.34 MessageCore . . . 27
4.1.35 Request . . . 27
4.1.36 Response . . . 27
4.1.37 XMLConverter . . . 27
4.2 Customer User Interface . . . 35
4.2.1 Introduction . . . 35
4.2.2 basic web pages . . . 35
4.3 Staff User Interface . . . 36
4.3.1 Introduction . . . 36
4.3.2 Reporter Interface . . . 36
4.3.3 Editor Interface . . . 36
4.3.4 Administrator Interface . . . 37

5 Diagrams/Drafts 38

6 Interface Prototype 41
6.1 HTML Prototype . . . 41
6.1.1 Introduction . . . 41
6.1.2 Design Principles . . . 41
6.2 WML Prototype . . . 41

6.2.1 Introduction 41
6.2.2 Design Principles 42

A Protocolls **43**
A.1 25.10.2001 43
A.2 31.10.2001 43
A.2.1 Answers to the open questions 44
A.3 07.11.2001 45
A.4 14.11.2001 46
A.4.1 Notices 46
A.5 21.11.2001 47
A.5.1 Notices 47

Chapter 1

Introduction

Information is a very important part of our life. Human beings always wanted to and want to be good informed on what is happening in their neighbourhood as well as in their cities, countries and continets. Fresh information has an influence on our decisions, and that is why people like to be fast and good informed. The most common way the people get informed today is by watching TV, reading newspapers, listening to the radio, etc. However all this information is to be collected and presented to the end-users. Computer-based system are very suitable for collecting and presenting of information.

1.1 Planning

1.2 Basic Organization

We are a group of 5 members. Each member has a certain task to perform. The tasks are determined through the role distribution. Each week there will be a session between the group members and a session between Mr Schweiger and the project leader. The project leader rotates from session to the session.The following dates for sessions are already set:

- Huskic Zenina on October 31, 2001
- ...

Each member of the group should prepare her/his own solution to the problem. These solutions will be worked through by the group. Group document contains parts of the single solutions. Project leader should present the group document to Mr Schweiger. The project leader makes a protocoll about this session. The protocolls are to be found as appendix to this paper.

1.3 Goal

Goal is to perform analysis of the system as good as possible in order to prepare a basis for future implementation. Further on, the members of the group should get a better understanding of the iteration process. The iteration process should result in an end document (project paper). This document should contain clear project definitions and requirements as well as implementation constraints.

1.4 Responsibilities

The project leader is Zenina Huskic. She is responsible for the overall organisation during the session. Michael Brachtl is responisble to watch the prepared time line. All other group members do not have any special responsibilities.

1.5 Problem Statement

The following text contains the theme and open questions to the theme. The questions are numbered and given as a footnote.
Eine Nachrichtenagentur ("e-News") stellt an uns die Anforderung den Informationssammlungsprozess und Informationspräsentationsprozess aufzubereiten mithilfe von EDV-technische Unterstützung (Projektname: "InfoWarrior"). Diese Prozesse sollten bewusst einfach [1] gehalten werden.
Als potenzielle Nutzer des Systems sollten natürlich Erfasser [2] [3] der Information [4] eingeplant werden plus Abrufer [5] von solchen Informationen. Die Abrufer von Informationen können einerseits Fremdsysteme [6] sein, andererseits Personen [7].
Daraus resultieren zwei mögliche Szenarien:

- Gesammelte Information wird von CodeWarrior [8] an andere Systeme weitergereicht (XML sollte dabei benutzt werden) [9]

- Gesammelte Information wird von CodeWarrior an Einzelpersonen weitergegeben [10]. Dies setzt natürlich eine entsprechende Aufbereitung [11] Information dar.

Als besondere Anforderung wird die einfache Wartbarkeit von CodeWarrior angemerkt, d.h. u.a. eine lokale Installation von Erfassungskomponenten und Darstellungskomponenten sollte nicht erfolgen [12]. Desweiteren sollte e-News die Möglichkeit haben, die Aufbereitung der Information für Einzelpersonen zu beeinflussen, ebenfalls ohne grossen Aufwand. Für die erste Phase von CodeWarrior wird eine Reduktion auf das Wesentliche [13] angestrebt aber mit den Optionen in weiteren Phasen Erweiterungen, in Kooperation mit unserer Gruppe, am System anzubringen.
Geplante Projektdauer ist bis Ende Jannuar 2002.

[1] What does "einfach = simple" mean, that is what is for customer simple enough?
[2] Who are Erfasser, "e-News" agency employees or we?
[3] How many data collectors are there?
[4] What is going to be prapared, that is, what is information content (text, pictures, multimedia)?
[5] Who are Abrufer, that is who can access the infomation, that is, are there different access levels?
[6] What kind of foreign systems?
[7] Which persons (e-News agency employees, we, or others)?
[8] What is exactly the name of the project, Infowarrior or CodeWarrior?
[9] Are there already other systems existing or not? If yes, does DTD for XML exist or not? That is, how is information delivered?
[10] How is information delivered to the end-users(per e-mail?)?
[11] How will information be prepared?
[12] Does it mean that this a web-based application , or what?
[13] What is for customer "wesentlich = important"?

1.6 First considerations(CRC Session)

In the following we give an overview of classes and few scenarios.
There probably may be many ways to select classes, we tried this method: Starting with the problem statement, we looked for nouns first as we considered them to be potential objects. As a next step, we thought about different scenarios to evaluate the suitability of our very first ideas and to enlarge the correspondence between concept and reality. Finally, we selected several questions that occured during the process. We hope many of them will be answered during this session in addition to the questions that are set as footnote(questons to the problem statement).

1.6.1 Identifying classes

- User
  - Data collector
  - Other users
    - Third parties
    - Foreign systems
- InfoWarrior-application
- E-News employees
- Information material
- E-News agency
- Access manager

Further on, the following canditate classes can be considered:

- Information Receiving Unit
- Information Storing Unit
- Request Receiving Unit
- Front End Unit
- System to System Unit

1.6.2 Some scenarios

- Can data collector XY manipulate all currently available data, and what happens if XY tries to access or change data out of scope of his access rights?
  Suggestion: There should be some kind of access manager that restricts access only to the field relative for collector XY. If XY tries to access date without permission there will be an error message shown.

- What happens if fresh news from some area is/are available?
  Suggestion: There are for example two possibilities:
  - User sets her/his preferences upon registration and gets news delivered per e-Mail automatically if there is something new in the area she/he is registered for.
  - User sets her/his preferences upon registration and when she/he visits homepage of e-news agency, upon login page is created with links to news that user is interested in.
  - others
- How does system respond if data collector XY enters data that already exist in database?
- What happens if user requieres news for which no fresh occurrence is available
  Sugestion:
  - System reports that no fresh news is available
  - System shows the most recent news, that user already knows about

Other scenarios that can be considered are:

- Enter News(manually)
- Delete News
- Edit existing News entries
- News request from automated system
- News browsing

Beyond these basic scenarios there can be some others, if user management is necessary, or we want to provide access control:

- Create User
- Login
- Modify User rights/preferences
- News entry not found
- Access violation

1.7 Open questions

- Can you specify the kind of sources from which information is likely to be received?
- Can you think of other scenarios you think are typical and/or important but not adequately represented in our first cut model?
- Do you have any aspects in mind that are completely missing in our concept?

Chapter 2

CRC Session

2.1 Problem Statement (Version two)

After the first session with the customer, the problem statement has been partly cleared.

- The news agency requires an information collecting and presenting software system (web-based application) that should be simple, which means data collecting, data presentation and meaningfull data input have to be simple organized.
- Data collectors are the reporters [1] employed at the e-News agency. Each reporter prepares at least one, and at most 5 news per day. The news are controlled by director. There are about 5 directors, and each director is in charge for approximately 50 reporters.
- Information should be accessed in two ways:
  - Pull: login
  - Push: via SMS, via e-Mail, ...

  There should be different access levels based on price model, that is:
  - free information available to the whole world
  - for very important, time dependent infromation one must pay in order to access it
- Information content should be all supported types (text, pictures, multimedia).
- Information will be distributed to other foreign systems and single persons. A foreign system is for example, the CNN page.
- There has to be a possibility for preparing information for each person (the preferences should be entered by the user upon registration).
- New information has to be faster available than the old one. Under "old information" is meant the information older than a month.
- In the first phase of this project the collecting and distributing process of information has to be clarified.

[1] Data collector = reporter

2.2 Open Questions

1. Which operating system is going to be used?
2. Who is responsible for hadware (who is going to buy hardware, who is going to maintain hardware, ...) ?
3. Which information is needed for registration?
4. Should a user be able to change her/his preferences at some point later on?

2.3 CRC Card Session

2.3.1 General

- CRC Cards
  CRC Cards are invented by Ward Cunningham and Kent Beck.They were, and are, concered about how to teach people that same deep knowledge of how to use Smalltalk that they had gained. From this question of how to teach objects came the simple technique of CRC (Class-Responsibility-Collaborataion) cards.
  Individual CRC Cards are used to represent objects. The class name is written at the top of the card, responsibilities are listed down the left side, collaborating classes are listed to the right of each responsibility.
  In CRC Session the developers will crowd around a table, each person picking up a card as they describe how the class participates in some use case.
- The Session
  The ideal group for a CRC card session is five or six people. It should be composed of developers, domain experts, and an object-oriented technology facilitator.
  Steps in a CRC card session:
  1. Creating class
     - Find all of the nouns and verbs in the problem statement
       The nouns are a good key to what classes are in the system, and the verbs show what their responsibilities are going to be.
     - Pass out cards and assign the class to the member of the group.
       Each person should be responsible for at least one class. Each person records the name of their class on a card. One class per card.
  2. Responsibilities
     - Add the responsibilities that are obvious from the problem statement or the name of the class. You don't need to find them at all or any. The scenarios will make them more obvious.
  3. Superclasses and Subclasses
     - Superclasses and subclasses can be defined at any time they become obvious.
  4. Attributes

  - Attributes of class are an implementation detail and are generally not defined at all until the design phase, but they can be defined at any time the group thinks it is appropriate.

5. Scenario execution
   - Scenarios are walk-throughs of the functions of the system in detail.
   - Start with scenarios that are part of the system's normal operation first, and then exceptional scenarios, like error recover, later.
   - Decide which class is responsible for this function. The owner of the class picks up his card. When a card is in the air it is an object and can do things.

2.4 InfoWarrior CRC Session

1. Creating class
   - Nouns
     - application
     - user interface
     - news
     - category
     - reporters
     - editors
     - access manager
     - users
     - external systems
     - storage
     - archive
   - Verbs
     - collect
     - distribute
     - input
     - send
     - login
     - control access
2. Some scenarios
   - Can data collector XY manipulate all currently available data, and what happens if XY tries to access or change data out of scope of his access rights?
     Suggestion: There should be some kind of access manager that restricts access only to the field relative for collector XY. If XY tries to access date without permission there will be an error message shown.

- What happens if fresh news from some area is/are available?
  Suggestion: There are for example two possibilities:
  - User sets her/his preferences upon registration and gets news delivered per e-Mail automatically if there is something new in the area she/he is registered for.
  - User sets her/his preferences upon registration and when she/he visits homepage of e-news agency, upon login page is created with links to news that user is interested in.
  - others
- How does system respond if data collector XY enters data that already exist in database?
- What happens if user requieres news for which no fresh occurrence is available
  Sugestion:
  - System reports that no fresh news is available
  - System shows the most recent news, that user already knows about
- Enter News(manually)
- Delete News
- Edit existing News entries
- News request from automated system
- News browsing
- Create User
- Login
- Modify User rights/preferences
- News entry not found
- Access violation

2.5 Essential Use Case Modeling

2.5.1 General

- Use Cases
  - "a behaviorally related sequence of transactions in a dialogue with the system",(Ivar Jacobson)
  - For example: Use Cases for a word processor might include building an index or inserting a picture.
- Use Case Diagrams
  - provide a way of describing the extarnal view of the system and its interactions with the outside world. In this representation the outside world is represented as actors.

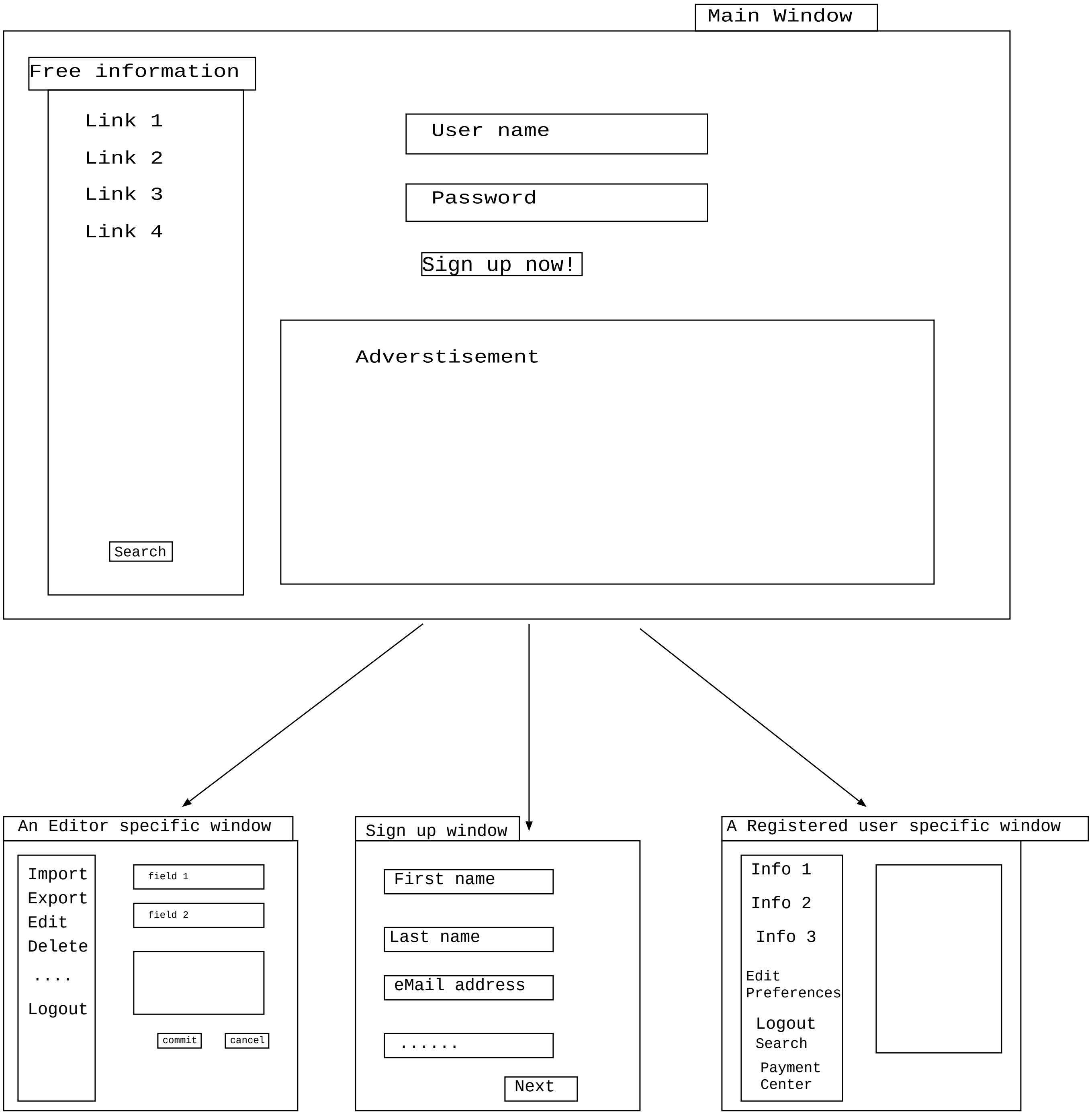
- Actors
  - Actors are roles played by various people, or other computer systems. One person may play many roles, and a role may have many people playing it. Use Cases are then typical interactions that the actor has with the system.
- Relationships
  - are represented through the lines between actors and Use Cases, and between Use Cases and Use Cases.

2.5.2 Use Cases in InfoWarrior

- Actors
  - User
  - Reporter
  - Editor
  - InfoWarrior
  - News
  - ...
- Relationships
  - Subscribe for News
  - Login
  - Change Preferences
  - Enter Information
  - Modify Information
  - Delete Information
  - Archieve Information
  - Send Information
  - ...

2.6 Essential User Interface Prototyping (First considerations)

System contains one main window with links/ buttons to the other pages/windows. It is the same both for normal users and for editors. Upon login new User Interface is shown with features available depending on access level.


Main Window
Free information
Link 1
Link 2
Link 3
Link 4
Search
User name
Password
Sign up now!
Adverstisement
An Editor specific window
Import
Export
Edit
Delete
....
Logout
field 1
field 2
commit
cancel
Sign up window
First name
Last name
eMail address
......
Next
A Registered user specific window
Info 1
Info 2
Info 3
Edit Preferences
Logout
Search
Payment Center

Chapter 3

Use Cases Description

3.1 Identifying Actors

- Editor
- Reporter
- End-user
- External System
- Administrator
- InfoWarrrior Application
- Access Control Manager
- News
- InfoWarrior Search Machine
- Storage Unit

3.2 Identifying Use Cases

- Subscribe to InfoWarrior Services
- Unsubscribe from InfoWarrior Services
- User logs in
- End-User edits Preferences
- Editor edits News
- User searches News
- Administrator modifies User Data

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 clicks "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 clicks "Unsubscribe From InfoWarrior Servceses" 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, end-user) 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, end-user) 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:** End-User edits Preferences
   Description: An End-User wants to edit her/his preferences. After login every user has a link "Edit Preferences". When she/he clicks 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 clicks a button "Submit changes" in order to apply changes to her/his account.

5. **Name:** Editor edits News
   Description: An editor wants to edit news. That means she/he wants to add, delete, or alter news. Upon login every editor has buttons for each of these actions. When she/he clicks 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, end-user) wants to search for recent news. She/he clicks "Search" link in order to start search page. A user enters wanted pattern in the corresponding field and clicks "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, end-user) wants to search for older news.

She/he clicks “Advanced Search” link in order to start the advanced search page. Afterwards she/he enters wanted pattern in the corresponding field and clicks “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 clicks search. The system returns a list of matching users, from which the administrator can select a user.

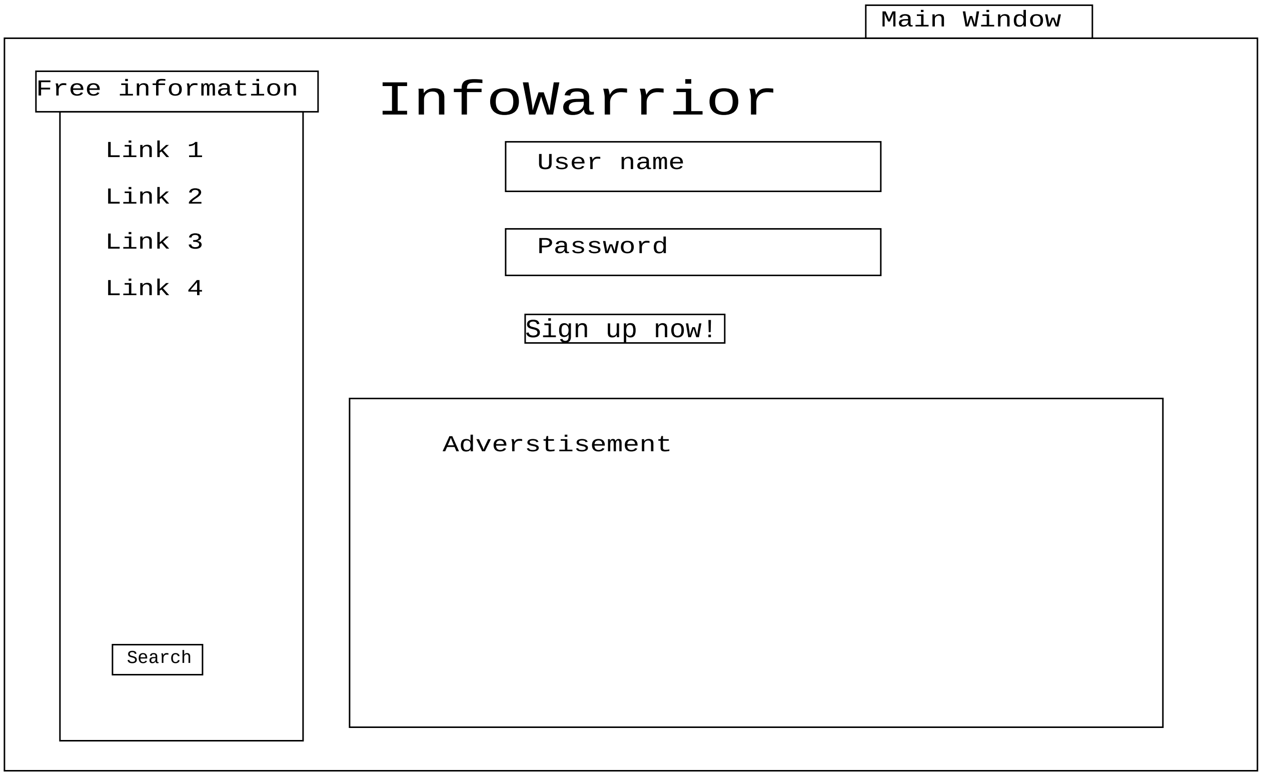
11. **Name:** Administrator modifies User Data
Description: The administrator wants to modify user data or privileges. He searches the specific user as described in “Administrator searches User”. He selects to modify the user and the user modification page is returned. There he changes the data and confirms 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.

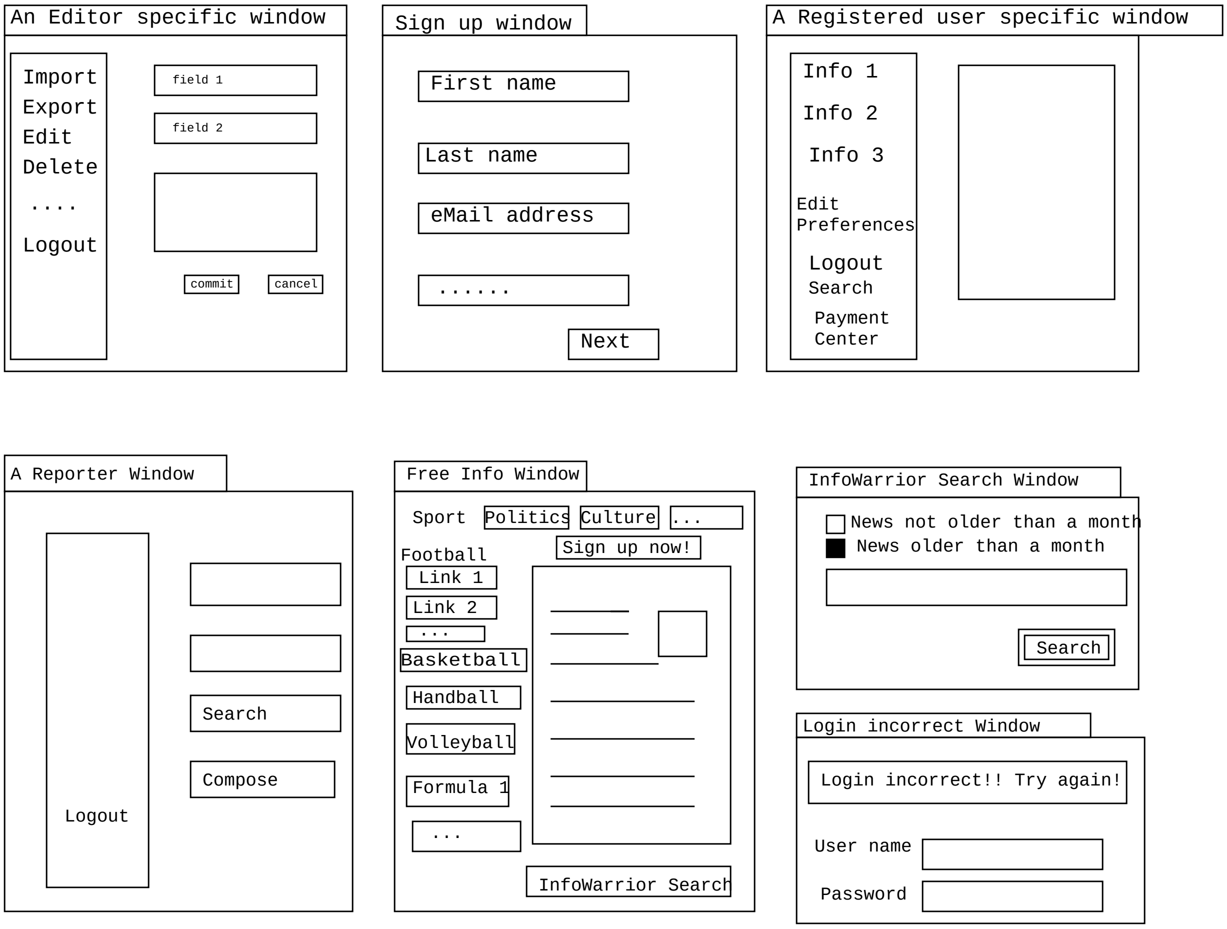
13. **Name:** Administrator creates a category
Description: The administrator wants to create a new category for news. He selects the link on the main page and enters the name of the category. After the administrator confirms the data, the system creates a new category

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.

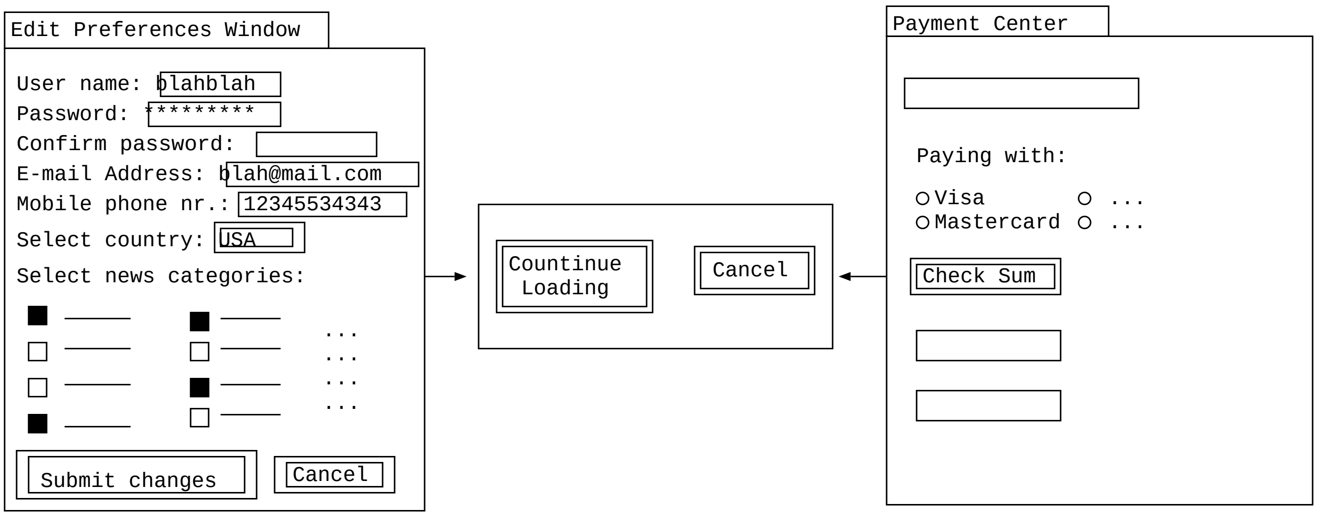
3.3 Essential User Interface Prototyping



System contains one main window with links/buttons to other pages/windows. It is the same for normal users, editors and reporters. Upon login new User Interface is shown with features available depending on access level and demanded service. It can be “An Editor specific Window”, “A Reporter specific Window”, “Sign up window”, “A registered User specific Window”, “InfoWarrior Search Window”, or “Free Info Window”



After successful login, a user can open different windows/pages, for example:



3.4 Open Questions (3)

1. Can an editor edit her/his preferences?
2. Can a reporter edit her/his preferences?

Chapter 4

Classes

4.1 Classes List

- Storage
- UserStorage
- Message Storage
- Primary Message Storage
- Secondary Message Storage
- User
- 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 storing purposes.
Public Operations:
Connect () : Boolean
Close () : Boolean
GetReport () : REPORT
GetSize () : Integer

4.1.2 UserStorage

Class for Users.
Public Operations:
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

Abstract class for storing messages.
Public Operations:
GetMessage (id) : MESSAGE
AddMessage (MESSAGE) : boolean
FindMessage (REQUEST) : RESPONSE

4.1.4 PrimaryMessageStorage

Class for storing current messages.
Public Operations:
UpdateMessage (MESSAGE) : boolean
DeleteMessage (id) : boolean AddCategory (CATEGORY) : boolean
UpadteCategory (CATEGORY) : boolean

4.1.5 SecondaryMessageStorage

Class for storing archived messages.

4.1.6 User

Abstract class for storing real person users of the system.
Public Attributes:
Name
Address
EMail
Phone
Mobile
CreditCard
Password
Subscriptions
LastLogin
Birthdate
Company
Operating System
Browser

Homepage
Login

4.1.7 Customer

Class for Customers of the System.
Public Attributes:

4.1.8 Reporter

Class for Roprters.
Public Attributes:
Department
Superior
Public Operations:
getArticles(): MESSAGES
getArticlesPending: MESSAGES

4.1.9 Editor

Class for editors of the system.

4.1.10 Editor

Class for editors of the system.

4.1.11 Administrator

Class for administrators of the system.

4.1.12 External System

Abstract class for external systems.

4.1.13 UserInterface

Abstract class for UserInterfaces.

4.1.14 CustomerInterface

Class for customers of the system.

4.1.15 ReporterInterface

Class for reporters.

4.1.16 EditorInterface

Class for editor.

4.1.17 AdministratorInterface

Class for administrators of the System.

4.1.18 SystemInterface

Class for access to systems by forgein systems.
Public Operations:
retrieveRequest(REQUEST): RESPONSE

4.1.19 Media Object

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

4.1.20 Picture

Class for Pictures.
Public Attributes:
Heigth
Width
Depth
Date
dpi

4.1.21 Video

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

4.1.22 Audio

Class for Audio files.
Public Attributes:
Length
SampleRate

4.1.23 Video

Class for Pictures.
Public Attributes:

4.1.24 Message

Class for messages.
Public Attributes:
Author
Title
Abstract
Text
Date
Editor
Category
PayLevel
LastChanged
Released

4.1.25 MessageSet

Class for messages.
Public Operations:
GetMessages():MESSAGES

4.1.26 Paylevel

Class for paylevels.
Public Attributes:
Name
Cost

4.1.27 Category

Class for categorys.
Public Attributes:
Name
Abstract

4.1.28 ACLSecuredObject

Abstract class for accesscontrolledlist Objects .

4.1.29 Secured Object

Interface for ACLSecuredObject.
Public Operations:
isAllowed (USER, ACTION) : boolean
GetParent()

4.1.30 ACL

Class for AccessControlledLists.
Public Operations:
GetPermission()
GetRoles()
HasPermission()

4.1.31 Permission

Class for permissions.
Public Attributes:
Name

4.1.32 Roles

Class for roles.
Public Attributes:
Name
Public Operations:
GetMembers(): USER
AddMembers(USER): boolean
RemoveMembers(USER): boolean
isMember(USER): boolean

4.1.33 UserCore

Class for user operations.
Public Operations:
executeRequest(REQUEST):RESPONSE

4.1.34 MessageCore

Class for messages.
Public Operations:
executeRequest(REQUEST):RESPONSE

4.1.35 Request

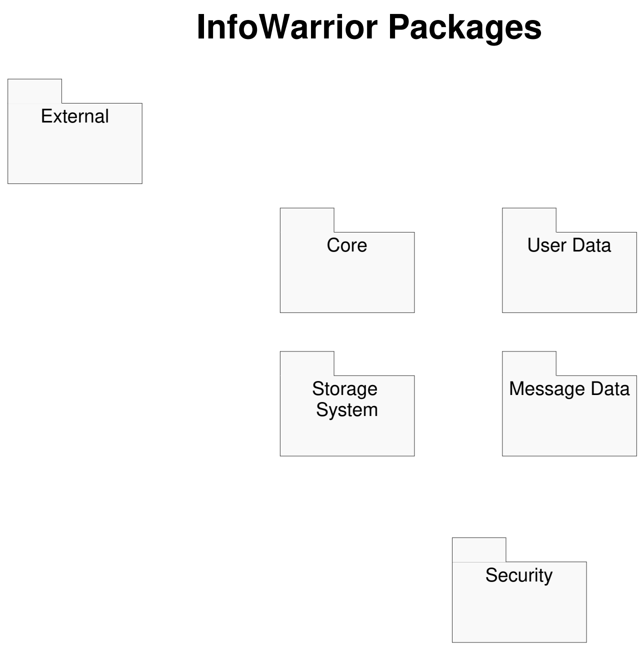
Class for requests.
Public Attributes:
Command

4.1.36 Response

Class for search results.
Public Attributes:
Errorcode
Results

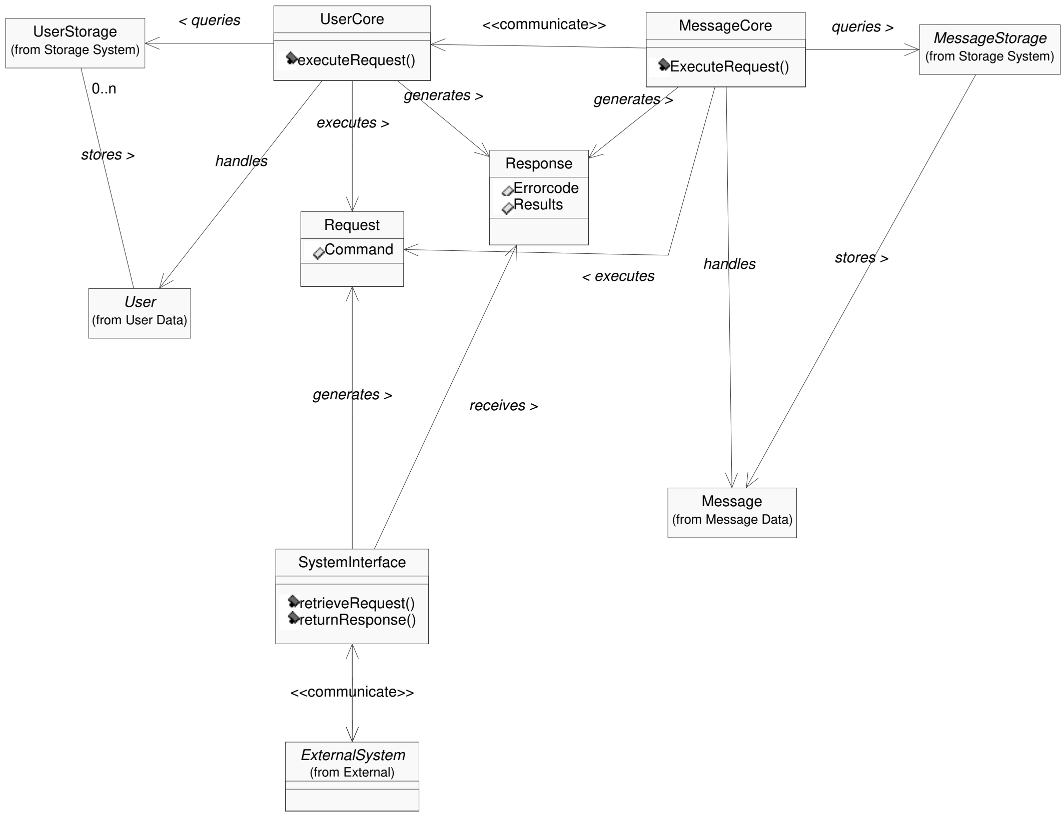
4.1.37 XMLConverter

Interface for converting XML to Strings and vice versa.
Public Operations:
XML2Request() Response2XML()



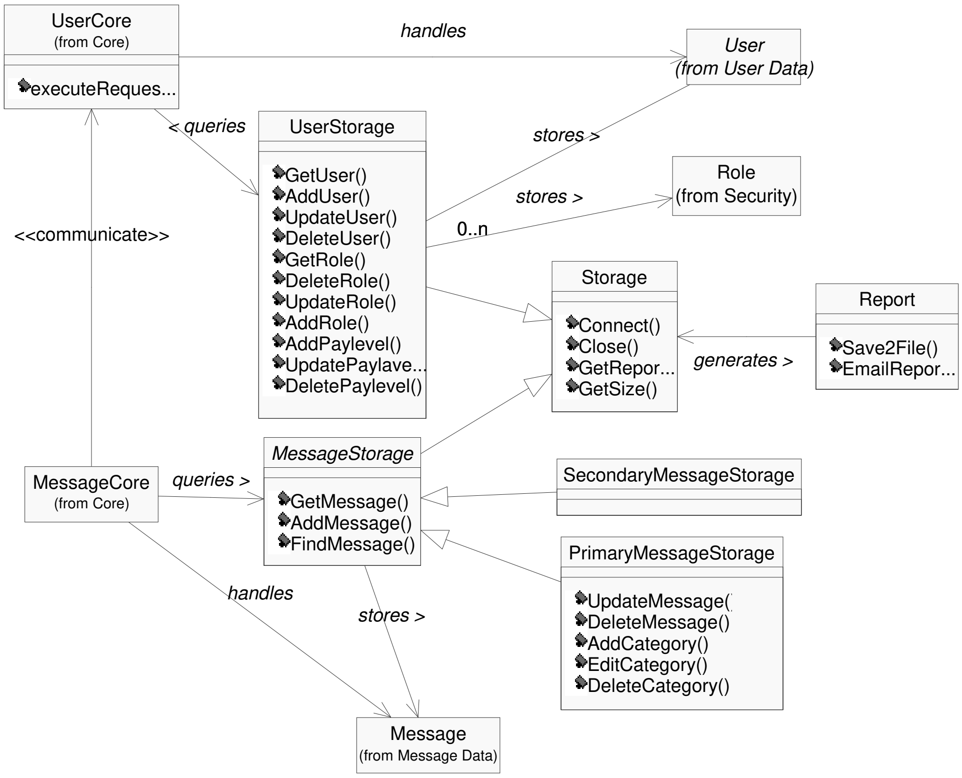
File: d:\users\albert\Class Diagram.mdl 16:12:41 Freitag, 30. November 2001 Class Diagram: Logical View / Main Page 1

Core Package



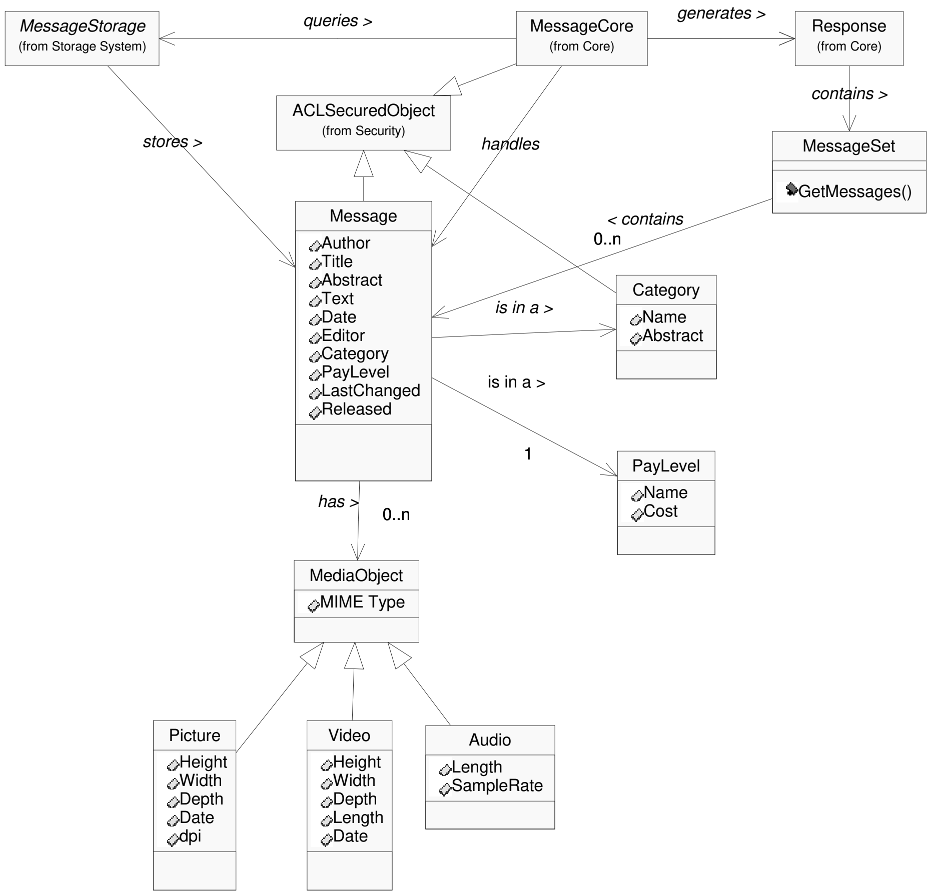
File: d:\users\albert\Class Diagram.mdl 16:14:25 Freitag, 30. November 2001 Class Diagram: Core / Core Page 1

Storage Package



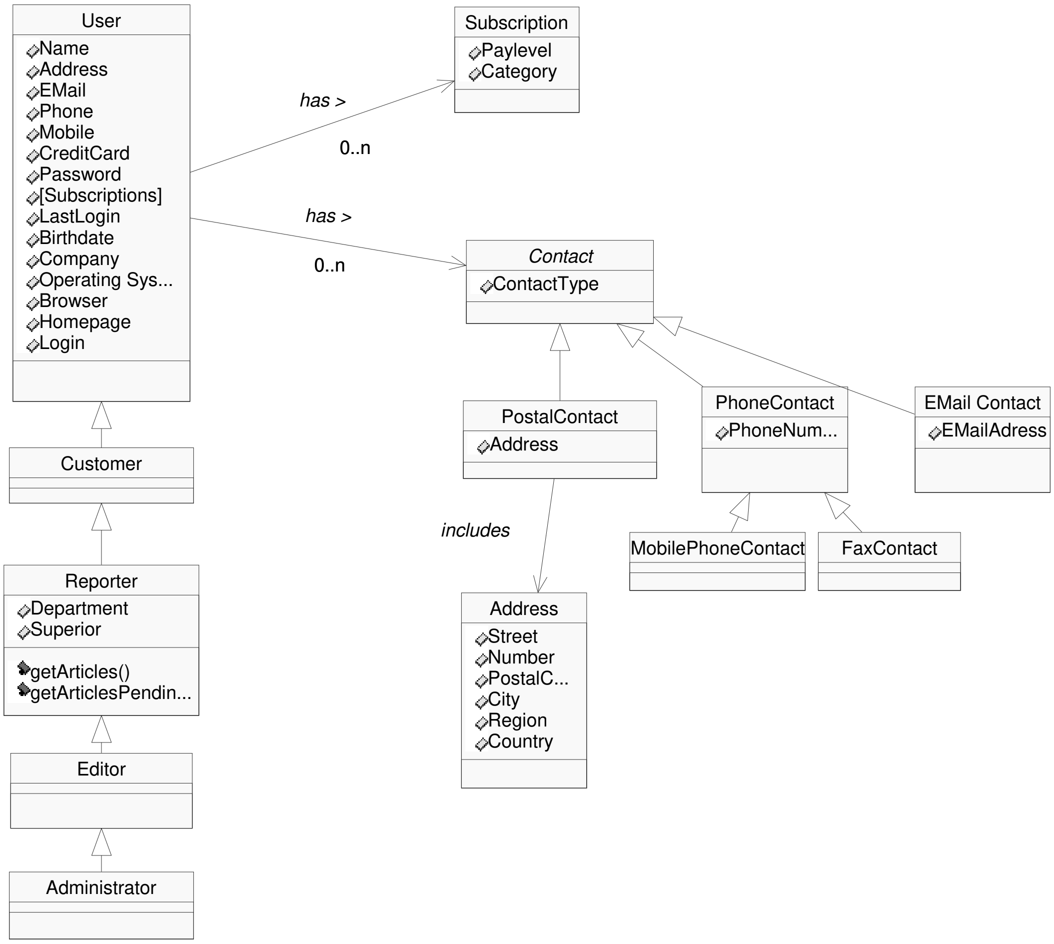
File: d:\users\albert\Class Diagram.mdl 16:18:52 Freitag, 30. November 2001 Class Diagram: Storage System / Storage Page 1

Message Package



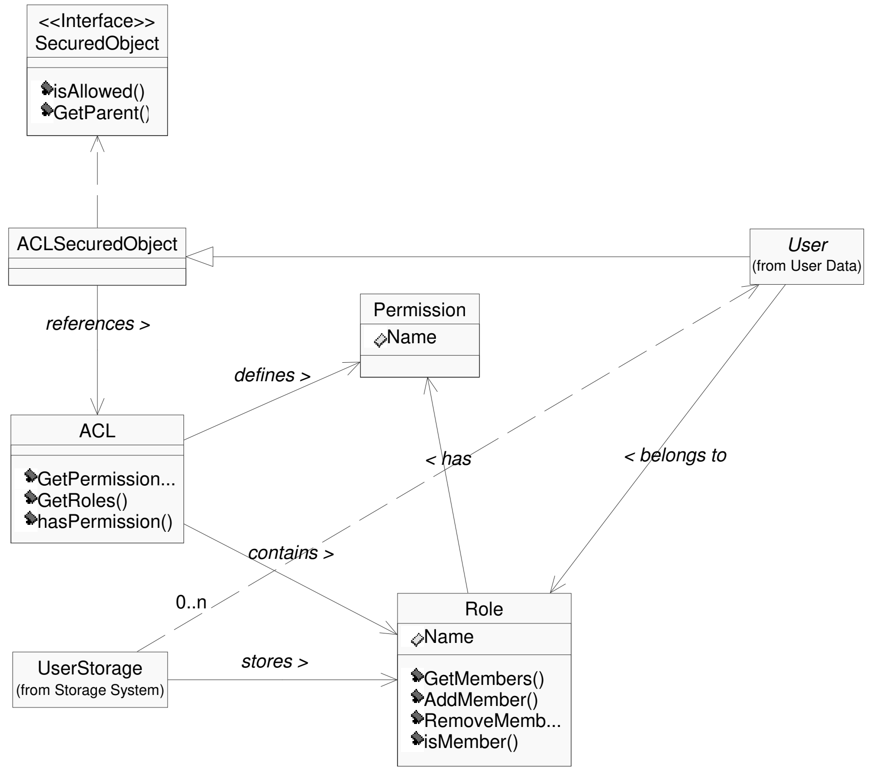
File: d:\users\albert\Class Diagram.mdl   16:16:28 Freitag, 30. November 2001   Class Diagram: Message Data / Message Data Page 1

User Package



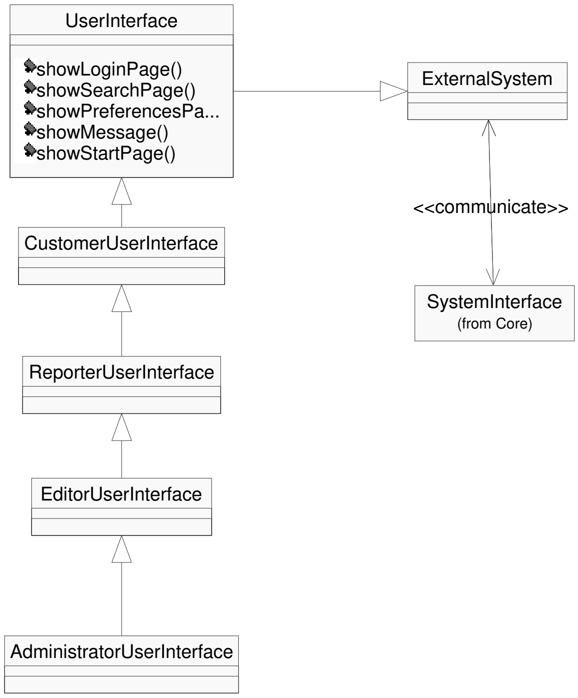
File: d:\users\albert\Class Diagram.mdl 16:19:59 Freitag, 30. November 2001 Class Diagram: User Data / User Data Page 1

Security Package



File: d:\users\albert\Class Diagram.mdl 16:18:15 Freitag, 30. November 2001 Class Diagram: Security / Security Page 1

External Package



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

4.2 Customer User Interface

4.2.1 Introduction

Since the costumer front-end is mainly web based, the user interface consists of web pages.

4.2.2 basic web pages

Start Page

the home page.
it is the first thing a (potential) costumer sees after typing in the url, so it should be attractive, clear and well-structured. It features the latest headlines and provides links to the most important subpages.
Important features: User Login, link to Sign Up Page.

Sign Up Page

main purpose: customer registration.
Features input boxes for user data, like name, surname, country, state, e-mail address, a login name and password etc, and options for pay level, payment mode.

Category Page

main purpose: structuring of articles, facilitating efficient browsing.
categories might be economy, science, politics, with subcategories world politics, regional politics etc. links lead to other category pages (related categories or subcategories), and article pages. Also provides an input box for quick search for articles within the category.

Search Page

provides advanced search options to fine-tune a search, like searching for older articles, AND/OR junction of keywords, searching within a specific category etc.
search results are displayed on Search Result Page.

Search Result Page

presents links to articles the search engine has found, also information about date, content, and category. may also show headlines of articels of a higher pay level than user's, with suggestions to sign up to higher level.

Welcome User Page

page that opens after user login, featuring headlines of user's preferred categories.

Preferences Page

main purpose: (registered) users can set different options like default search settings, preferred categories etc and subscribe to services that are within their pay level.

Article Page

Displays article text in attractive, readable form as well as pictures, offers accompaning multimedia files for download etc. Links to related articles.

4.3 Staff User Interface

4.3.1 Introduction

As the Staff User has a different approach to the InfoWarrior system than the customer, the requirements to a user interface are also quite different. The Staff User needs an interface to get his/her work efficiently done. If the Staff User Interface will be partiallly or fully web based like the Customer Interface remains open at this stage of design. The login procedure could be done via the Customer Interface Start Page.

4.3.2 Reporter Interface

Reporter Page

Each reporter's home page / main window, that is opened after login.
It gives an overview on the reporter's recent work, listing article headlines with information about date and whether the article has yet been submitted and released. Links to Edit Article Page (to create/edit articels), a task list, a search page/window to search for articles, and maybe an emailprogram to send email to the responsible editor.

Edit Article

enables reporters to compose, edit, and save articles, add files that accompany the article text (like pictures, sound files, and other multimedia), and submit them to the responsible editor. Input boxes for text, abstract and headline.

4.3.3 Editor Interface

Since editors can also write articles, their interface provides the same functionality as the reporters', with additions.

Editor Page

Home page / main window. Similar to the Reporter Interface, it displays recent article headlines from the editor's section, with information about date, author, and whether the article was yet released. Article headlines link to View Article Page/Window. Link to email program to send tasks to reporters.

View Article

Displays articles as they are seen on the homepage/Customer Interface (ie Article Page). Buttons to release, delete or edit article (-¿ Edit Article Page).

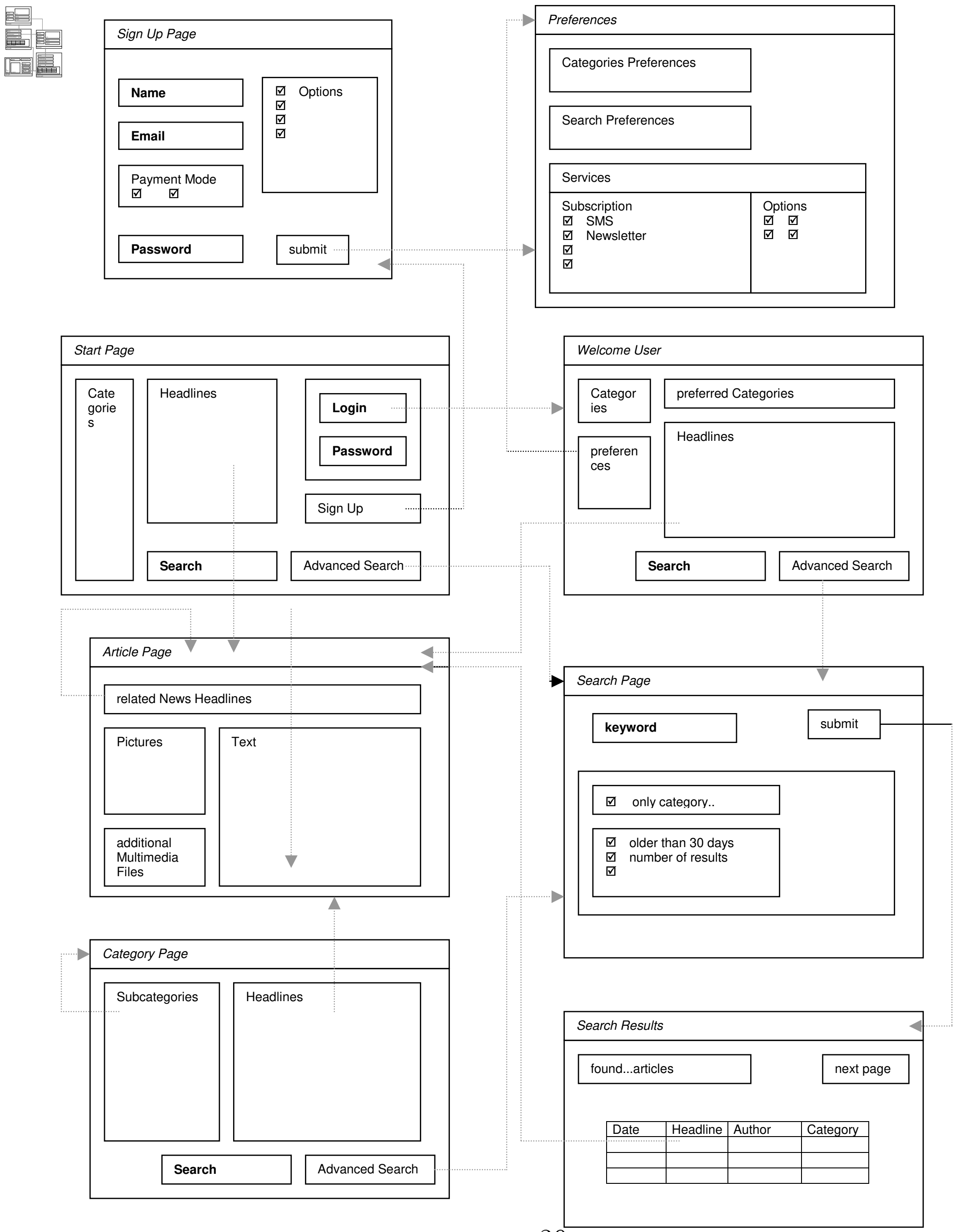
Edit Article

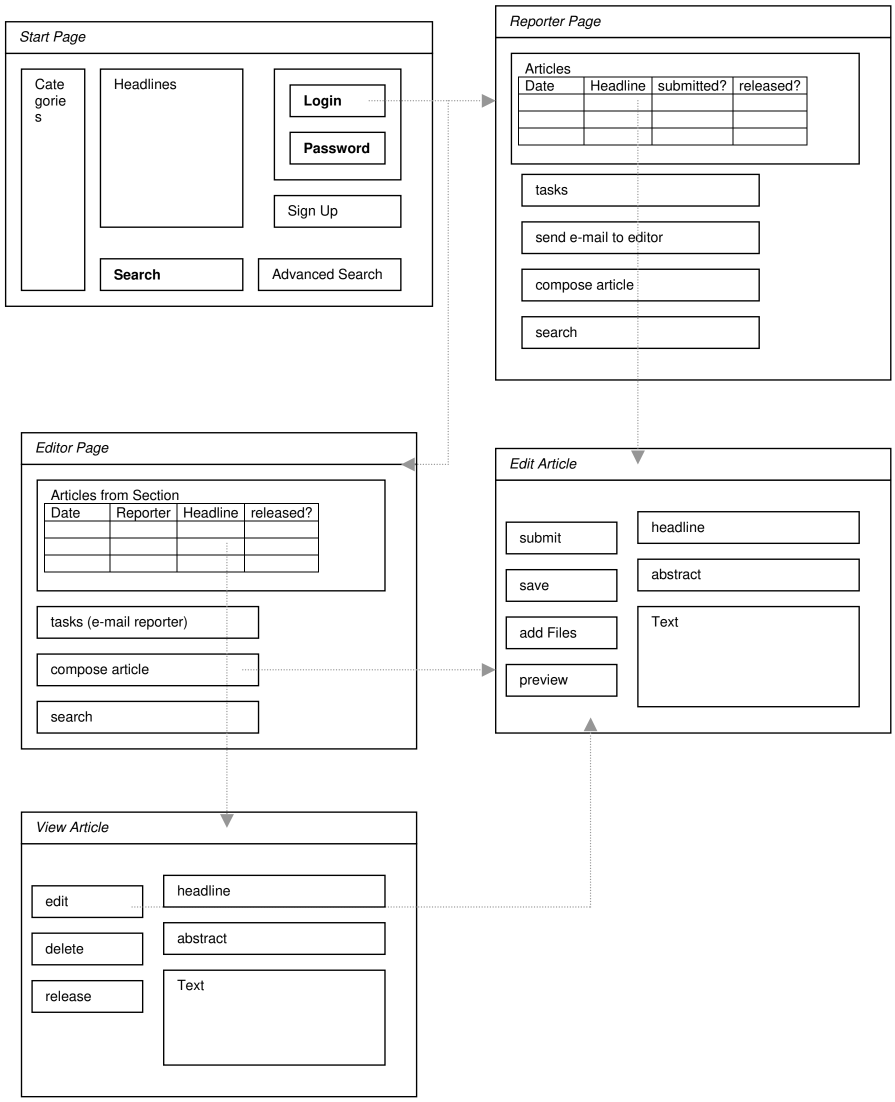
As in Reporter Interface. (See above)

4.3.4 Administrator Interface

Chapter 5

Diagrams/Drafts




Start Page
Cate gorie s
Headlines
Login
Password
Sign Up
Search
Advanced Search
Reporter Page
Articles
Date
Headline
submitted?
released?
tasks
send e-mail to editor
compose article
search
Editor Page
Articles from Section
Date
Reporter
Headline
released?
tasks (e-mail reporter)
compose article
search
Edit Article
submit
save
add Files
preview
headline
abstract
Text
View Article
edit
delete
release
headline
abstract
Text

Chapter 6

Interface Prototype

6.1 HTML Prototype

6.1.1 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.

6.1.2 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.

6.2 WML Prototype

6.2.1 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.

6.2.2 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
- 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

Appendix A

Protocolls

A.1 25.10.2001

1. Person responsible for protocoll: Huskic Zenina
2. Participating persons: Schweiger Roland, all students visiting Software Engineering PS course in WS 2001/02
3. Topics talked about: introduction, general information about PS (tasks, marks, presentation)
4. New assignments:
    - CRC Session
    - planning
    - role distribution
    - setting the goal
    - open questions

A.2 31.10.2001

1. Person responsible for protocoll: Huskic Zenina
2. Participating persons: Schweiger Roland, Theresa Krichhammer, Peter Gruber, Albert Meixner, Michael Brachtl and Huskic Zenina.
3. Topics talked about: Open questions
4. New assignments:
    - Aufgabe 3 (07.11): Verfeinern Sie die Problembeschreibung mithilfe der ersten CRC Session und der darin entstandenen Protokolle. Planen Sie eine weitere CRC Session um aufgetretene Fragen nochmals mit dem Kundenverteter zu besprechen. In dieser zweiten Session werden Use Cases eine Rolle spielen. Desweiteren sollten Sie sich Gedanken zum Essential User Interface Prototyping

machen. Das Verständins für die Problemdomäne ist bereits vertieft. In den Einzel- und Gruppendokumenten werden bis zum 07. 11 die Protokolle der ersten Session mitaufgenommen (Gruppenprotokoll in Gruppendokument und Einzelprotokolle in die Einzeldokumente). Dokumentieren Sie den Ausgang der ersten Session.

A.2.1 Answers to the open questions

1. What does "einfach = simple" mean, that is what is for customer simple enough?
   Simple means: data collecting, data presentation and meaningfull data input.

2. Who are Erfasser, "e-News" agency employees or we? How many data collectors are there?
   Reporters will prepare the news. The director controlls and issues the news. There are about 5 directors. Each director is responsible for approximately 50 reporters. Each reporter prepares at least one, and at most 5 news per day.

3. What is going to be prapared, that is, what is information content (text, pictures, multimedia)?
   Information content: text, pictures, multimedia...

4. Who are Abrufer, that is who can access the infomation, that is, are there different access levels?
   There are different access levels based on the price model:
   - free information available to the whole world
   - very important time dependent information will be paid

5. What kind of foreign systems?
   Foreign systems are: other pages (e.g. CNN page), SMS,...

6. Are there already other systems existing or not? If yes, does DTD for XML exist or not? That is, how is information delivered?
   There is no predefined DTD for XML.

7. How is information delivered to the end-users(per e-mail?)?
   Information to the end-users will be delivered in two ways:
   - push: via SMS, or via e-mail
   - pull: login

8. Does it mean that this a web-based application , or what?
   Yes, this is a web-based application.

9. What is for customer "wesentlich = important"?
   Important: collect and distribute important information
   Not important: fees accounting, customer accounting, etc.

10. Access time?
    The new information (not older than a month) should be faster available than the old one.

A.3 07.11.2001

Nov. 7, 2001

As a result of this second session, several more details of system requirements and administrative business were discussed and clarified:

- *Payment-Methods:*

  Invoice for payment concerning services that do not come for free will be done on a monthly basis. This should give a possibility to check whether users are still paying or willing to pay in time. Usually, invoices will be sent to users' postal adresses.

- The system should be developed as platform independent. The main aspect should be an easy maintenance of the system.

- *Actors:*

  The following items were agreed upon:

  - *human actors:* editors, reporters, users of the system (clients)
  - *external systems*
  - *archives:* containing news that reach further back in time
  - *storage units:* subdivided into a message storage (containing high priority information) and user data (adaministrating access control etc.)

- *entry page - user interface:*

  The entry page should present at least some accurate and on time information from all "over the world". Informations should refer to certain categories. Access to informations will be granted for different levels of payment. The user interface may also present an option to subscribe for services.

- *Sign-in-Procedures:*

  Sign-in is required for push-methods. Therefore, it will be necessary to extract informations from articles for different purposes. This should enable e.g. an SMS-News-Service, that is a minimum information in a given format.

- *Technical Requirements:*

  Editing of articles is done only once for different services. The input form of articles will thus provide seperate areas for entries such as headline, abstract (up to ... characters), categorization, payment-level of information, ... As a last instance, the editor will supervise all entries.
  Editors may compose news, too. Anyway, the two-level-process of releasement has to be secured, so any time an editor acts as "reporter", his article has to be approved by another editor.

- *Technical Aspects of Administration:*

  New subscriptions to services should be processed automatically. Thus, the system has to meet any requirements necessary. However, anytime an action concerning users of the system is taken, the administrator or people of the administration office should be informed personally. Frequent actions may be summarized as reports. Persons, however, should not be required to take actions in order to complete a standardized process.

In this second session, a first concept of User-Interface was presented and approved in general.

The following items should be prepared for the next meeting:

- *Use Cases:* The whole process should be described in natural language.
  For the most important Use Cases, drawings in UML may be prepared, too.

- In addition, transparencies concerning Use Cases, UML and User-Interface may be provided.

A.4 14.11.2001

1. Person responsible for protocol: Huskic Zenina

2. Participating persons: Roland Schweiger, Theresa Krichhammer, Peter Gruber, Albert Meixner, Michael Brachtl and Zenina Huskic.

3. Group Document presented by: Peter Gruber.

4. Topics talked about: Use Cases and Essential User Interface Prototyping in InfoWarrior.

5. New assignments (21. 11):

   - In dieser Aufgabe erstellen Sie ein Analysis Klassendiagrammes. Mit dem Essential User Interface, den CRC Cards und den Use Cases haben wir dafür das Fundament gelegt. Sie werden Ihre Ergebnisse einerseits in Ihren Papers dokumentieren, andererseits auf Overhead Folien eine Präsentation über diese Themen vorbereiten.

6. Project Leader (14. 11 - 21. 11) : Michael Brachtl.

A.4.1 Notices

- Reporter has to have a possibility to interrupt submitting of news ("submitted news" back to "not submitted news".)

- What kind of search will be available?
  There are two possibilities: keyword search or full text search.
  Team has to discuss this point.

- There is no deleting of data from database.

- We have to take care of naming convention (e.g., click vs. select).

A.5 21.11.2001

1. Person responsible for protocol: Zenina Huskic.
2. Participating persons: Roland Schweiger, Theresa Krichhammer, Peter Gruber, Albert Meixner, Michael Brachtl and Zenina Huskic.
3. Group Document presented by: Michael Brachtl.
4. Topics talked about: Class Diagram in InfoWarrior.
5. New assignments (05. 12):
   - In dieser Aufgabe erstellen Sie Prototypen für User Interfaces. Ihr Essential User Interface mit Use Cases und dem ausgearbeiteten Klassendiagramm sollte bzgl. Anwendbarkeit geprüft werden für je einen HTML und einen WML UI Prototypen. Aus diesen Prototypen werden Sie Rückschlüsse auf Essential User Interface Flow Diagrams ziehen und weitere Anforderungen an das System entdecken. Sie werden Ihre Ergebnisse einerseits in Ihren Papers dokumentieren, andererseits mithilfe von Programmen eine Präsentation über diese Themen vorbereiten.
6. Project Leader (21. 11 - 05. 12) : Albert Meixner.

A.5.1 Notices

- Our class diagram is to detailed, and the goal of a class diagram is to show basic classes and relationships. The implementation details are not important in this phase (Problem Domain Modeling.)
- UML Notation should be used.