Event Planner
Development of an event planning tool

Borislav Maric
Juni 2007

Examensarbete, 10 poäng, C
Datavetenskap

Datavetenskapliga programmet
Examinator/handledare: Fredrik Bökman
Medbedömare: Carina Petersson
Event Planner
Development of an event planning tool

by

Borislav Maric

Institutionen för matematik, natur- och datavetenskap
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:
borislavmaric@boro.org

ABSTRACT

The event planning tool described here is a software developed in the form of a web application enabling continuous access to the information about the events for persons interested in organizing events and those interested in participating at them. For event organizers the tool gives the opportunity to create, edit, present the events and to view the participation statistics for the events they are organizing. The software was developed with PHP and MySQL. The iterative procedure of its development is described in this document.
## Contents

1 Introduction ................................................................................. 5  
   1.1 Purpose .............................................................................. 5  
   1.2 Structure of the Report ...................................................... 5  

2 Background .............................................................................. 6  
   2.1 Unified Process ................................................................. 6  
   2.2 Object-Oriented Analysis and Design- OOA and OOD .......... 6  
      2.2.1 Use cases .................................................................. 6  
      2.2.2 Diagrams used in unified modeling language (UML) .... 6  
      2.2.3 Web application development .................................... 6  
      2.2.4 Web programming languages and scripts .................... 7  
      2.2.5 Database management ................................................ 7  
      2.2.6 MySQL ..................................................................... 7  
      2.2.7 PHP ........................................................................ 7  
      2.2.8 AJAX ....................................................................... 7  

3 Method ..................................................................................... 7  

4 Inception .................................................................................. 9  
   4.1 Event Planning ................................................................. 9  
   4.2 Liquidice .......................................................................... 9  

5 Elaboration's iteration 1 ............................................................. 10  
   5.1 Analysis .......................................................................... 10  
      5.1.1 Prerequisites and Requirements ................................. 10  
      5.1.2 Software requirements specification ......................... 10  
      5.1.3 Use Cases ................................................................. 10  
   5.2 Design ............................................................................. 11  
   5.3 Implementation ................................................................ 11  
      5.4 Test ............................................................................. 11  
      5.4.1 Verification ............................................................... 11  
      5.4.2 Validation ................................................................ 12  
      5.4.3 Screenshots ............................................................... 12  

6 Elaboration's iteration 2 ............................................................. 12  
   6.1 Analysis .......................................................................... 12  
      6.1.1 Prerequisites and Requirements ................................. 12  
      6.1.2 Software requirements specification ......................... 12  
      6.1.3 Use Cases ................................................................. 12  
      6.1.4 UML ........................................................................ 12  
   6.2 Design ............................................................................. 13  
   6.3 Implementation ................................................................ 13  
      6.4 Test ............................................................................. 13  
      6.4.1 Verification ............................................................... 13  
      6.4.2 Validation ................................................................ 13  
      6.4.3 Screenshots ............................................................... 13  

7 Construction's - Iteration 1 ....................................................... 14  
   7.1 Analysis .......................................................................... 14  
      7.1.1 Prerequisites and Requirements ................................. 14  
      7.1.2 Software requirements specification ......................... 14  
      7.1.3 Use Cases ................................................................. 14  
   7.2 Design ............................................................................. 14  
   7.3 Implementation ................................................................ 14  
      7.4 Test ............................................................................. 14  
      7.4.1 Verification ............................................................... 14  
      7.4.2 Validation ................................................................ 15  
      7.4.3 Screenshots ............................................................... 15  

8 Construction's – Iteration 2 ....................................................... 15  
   8.1 Analysis .......................................................................... 15  
      8.1.1 Prerequisites and Requirements ................................. 15  
      8.1.2 Software requirements specification ......................... 15
1 Introduction

Most of the people have organized some kind of events if not “Olympic games” then some celebrations as e.g. “New Year’s Eve”. In fact every meeting can be an event even though only two persons are involved. In this writing we discuss the events where many people are involved.

When many people are involved the most important intentions should be concentrated on the information availability and validity. Information must be continuously accessible if possible and it has to be ensured that it is received by most of event participators in order to ensure that as few people as possible try to attend an event in case the event is cancelled. The Internet makes it possible to access information at any time a user wishes and to have access to it. Because of that reason, information technology is a very efficient way to inform the event participators about the status of an event but other traditional ways of communication as letter, telephone and fax should not be ignored as well.

Event organizers can use information technology to store information about an event and make that information available for others throughout web applications. Invited persons interested in participation at an event can by the other functions e.g. register themselves by using those web applications. The number of participators can be monitored by statistics for an event.

The intention during the project described in this writing was to make an application that would contain as many of the functions that could be used for an event planning as it was possible to make during the project’s time limitation. The functions, which were planned but were not implemented, are also listed and some that would be useful to implement are also mentioned.

1.1 Purpose

The purpose of the project was to develop a “web application” [1] for storing, retrieving and manipulating data needed for event planning (see http://www.xtevent.com/), is described, discussed and documented in this writing. The project was agreed with the organization “Liquidicce”, which announced its project in the “Exjobb-poolen” (see enclosure 1).

The purpose of the “Event planner” application is to register potential customers, to give them opportunity to create, edit and publish events and to present events to the persons interested in the participation in certain events. Beside that, the system contains functionalities for sending messages to the forums, for sending e-mails and for registration of participation for certain events.

The potential customers are expected to be the owners of different consulting companies and the locations for conferencing as well as organizers of meetings, celebrations and similar events.

1.2 Structure of the Report

The web application is developed as a “database website” [2] application in PHP and MySQL. A brief description of MySQL database and PHP used for application development is given under the heading “Background”.

Under the heading “Method” the unified process (UP) used during this project as “iterative software development process” [3] is described.
The system requirements, use cases and UML diagrams and results from the verification, validation of the web application are made and presented together with the web application under the headings “Inception”, “Elaboration”, “Construction” and “Transition”. Iterations of each phase are described under the subheadings “Analysis”, “Design”, “Implementation” and “Test”.

Under the heading “Discussion” the problems, which emerged during the web application development, are discussed and the appropriate solutions suggested.

The heading “Conclusion” consists of the summary of the project management.

Under the heading “References” information about the books, documents and the links used as templates and references are listed.

Documents made and collected throughout the project development are attached under the heading “Enclosure”.

2 Background

UP and the OOA/D object-oriented analysis and design was used during the development of the project with the Craig Larman book, “Applying UML and Patterns- An Introduction to Object-Oriented Analysis and Design and Iterative Development”, as a reference book.

2.1 Unified Process

The unified process- UP [4] is an iterative approach to software development where the project development usually is divided into several iterations, each consisting of the analysis, design, implementation and testing disciplines across the inception, elaboration, construction and transition phases. That means that each of the phases of the UP is divided into several iterations which are differing from each other in the time of execution and in the time dedicated for each of the UP disciplines: requirements analysis, design, implementation and testing.

2.2 Object-Oriented Analysis and Design- OOA and OOD

Object-oriented analysis and design is the object-oriented approach to the software development throughout the UP, which uses the Unified modeling design- UML and the object-oriented programming languages for development of software applications [5].

2.2.1 Use cases


2.2.2 Diagrams used in unified modeling language (UML)

To ensure that the methods for modeling of diagrams with the UML are standardized, the “ISO standardized version of the Unified Modeling Language (UML, v1.4.2)” document [8] was used because that is the last version of the UML standardized by ISO.

2.2.3 Web application development

As a system requirement, in the project description is listed that the application should be a web-based tool, which implies the development of a web application (see enclosure 1).
2.2.4 Web programming languages and scripts
As server-side programming language the PHP web programming language was used, as Liquidice required it. On the recommendation of the University of Gävle, the AJAX scripting language was aimed to be used as client-side scripting language.

2.2.5 Database management
Under the project description the use of the MySQL database is also listed as a system requirement (see enclosure 1). For database access API in PHP was used, too.

2.2.6 MySQL
The MySQL database is an “open source” [9] database founded by David Axmark, Allan Larsson and Michael Widenius and is registered under the Free Software Foundation” [10] owned by the MySQL AB company [11]. For the Event Planner application the version MySQL 4.0. was used because that version is going to be used by Liquidice.

2.2.7 PHP
The PHP [12] stands for Hypertext Preprocessor. PHP is an interpreted language initially written by Rasmus Lerdorf and later rewritten to the present Zend Engine by Zeev Suraski and Andi Gutmans and released under the Free Software Foundation. The first PHP version with object-oriented features was added in PHP3 while in PHP5 the object handling was rewritten [13]. This event planner application was developed for the version PHP4 and later.

2.2.8 AJAX
AJAX is the abbreviation for “Asynchronous JavaScript and XML” and is used for the development of interactive web sites. The communication between AJAX and server is maintained with API called “XMLHttpRequest objects” that can be used by JavaScript [14]. In this project AJAX was used for searching information stored in the database tables under certain categories.

3 Method
To ensure the standard norms and qualities of application development, the iterative approach through phases of unified process (UP): inception, elaboration, construction and transition and its disciplines: analysis (business modeling and requirements), design, implementation and test, was used to develop the system project [15]. Most attention was set on the fulfilling the basic idea of that approach, which is that “each iteration includes its own requirements analysis, design, implementation and testing activities” [16].

During the inception phase the initial requirements, use cases and a rough description of the possible course of the project development were described.

In the elaboration phase the project description was redefined, the core architecture and some implementation of the so-called “high-risk” elements of the application were done and validated.

During the construction phase the low-risk elements of the application were made and in each of the phase’s iterations it was aimed to develop and test the executable updates of the software application.
Test of what had been made in the previous iterations of the UP was done during the transition phase and its deployment on the users was tried to be made. The further suggestions for the application modification given by the users were considered and applied [17].

During the iteration of each phase verification of the application was made which means inspected if the application was made in the right way [18]. The check-list from the Patti Coons, “Gala-The special Event Planner for Professionals and Volunteers” with the description of the proper steps within the event planning relevant for this project development was used for its verification [19].

Validation was also made and that means controlling if the right application was made according to the software requirement description in each of the UP phases. Liquidice made the validation of the application to approve its worth and significance for Liquidice and its potential customers [20].

The requirements for the project development were distinguished and discussed in collaboration with Liquidice and the university’s assistant throughout all phases of the project and were analyzed in the beginning of each iteration for further development of the project.

During the inception phase of the project, the structure of the report was made and a brief description of the project formulated. In the beginning of the first iteration of the elaboration phase the list of requirements for the software functionality created by Liquidice was further analyzed with their assistance. In the first iteration of the construction phase the prototype application was implemented for the validation.

To ensure that the methods for modeling of diagrams with the UML are standardized, the “ISO standardized version of the Unified Modeling Language (UML, v1.4.2)” document was used [21] as that one, until now, is the only standardized version of the UML.

As a system requirement in the project description is listed, that the application should be a web-based tool, which implies the development of a web application (see enclosure 1).

As server-side programming language for this web application development, the PHP web programming language was used, as Liquidice required it. To ensure the application to be executable on any kind of browser, the server-side application in PHP was developed first. After that, the client-side script with JavaScript and AJAX was added to the application to optimize its performance capacity.

For this application development the PHP version 4.3.10-19 was used because that version is the one used by Liquidice. Initially the application had been developed under the PHP5 environment but it was found necessary to adapt it to the 4.3.10-19 version, which required more time than planned for the implementation of the application. That had consequences also on the validation of the application, which was often prolonged. PHP functions were used for database access too. More information about the PHP interpreted language can be found on the PHP’s homepage [22].

The application was developed on Macintosh OS X and “Darwin kernel version 8.6.2” with “Apache/2.0.59(Unix) PHP/4.4.4” and “PHP/5.1.6”. Tests were done on two web servers: “tatooine 2.6.11.12” with “Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-20” running on Debian Linux OS and “Zeus_Server 2.4.31-7” with “Apache/1.3.37 (Unix) PHP/4.4.7” running on Trustix Secure Linux OS.
Under the project description the use of the MySQL database (see enclosure 1) is also listed as a system requirement. For the Event Planner application the version MySQL 4.0.24 was used as the version available on the Liquidice’s web server.

The AJAX was used in the file “addressBook.php” for searching the persons from the contact list and JavaScript in “registerUser.php” and “createEvent.php” for verification of inserted data.

The final application is developed with most of functions executing on satisfactory level. To ensure that information is going to be received was second priority and because of that it is not developed during this project because of the time limitation. On Liquidice’s request some functions e.g. “Browse events” were developed but later their function was reduced to registered users only.

4 Inception

The inception phase was started with the research of the business domain in the planning of the events. It was found that event planning is characterized by the importance of the information-oriented application in contrast to data-oriented applications. That means that it is important that the data collected in the database is also comprehended by the users and not only accessible for them. An event is actual only if its participants are ensured that the event is going to take place. That can be ensured only by the continuous collaboration between the participators and the organizer. The event is not taking place if a certain number of persons are not going to attend it, which differs a lot from the usual booking systems where the future action of one user of the system does not influence on the other users. Because of that reason the event forum’s interface and the implementation of the functions for sending messages is very important for the application’s validity. The organizers of events are going to find its e-mail functions very useful even if they already have an e-mail account. The e-mail contains a function that ensures that the letter is going to be sent next time the “send e-mail” function is used, when it could not be sent in the first try. The e-mail gives also the opportunity for users to send e-mails in bulk.

4.1 Event Planning

For the organization of an event it is necessary to have the information about the event accessible for each person involved in its organization. A very good way to achieve that is to use web sites where all updated information about a certain event can be presented and accessed at any time. Invited persons interested just in participating in a certain event can apply there for the participation in that event and have information available when it is updated.

4.2 Liquidice

Liquidice is a consulting company helping the small business oriented firms to establish and market their business. The business strategy of Liquidice is to use a flexible model for receiving revenues as a set of monthly fees, get paid per event and per attended bookings and advertisement made throughout services they offer to their customers and partners.
5 Elaboration’s iteration 1

5.1 Analysis

Throughout the brief analysis and the first iteration of the UP and in the inception phase of the project development it was found, that in order to ensure the system security, it is important to use the php function when the user requests the data to be stored or displayed and not AJAX. The AJAX can be used for database access at the time the user is logged in.

Throughout the analysis it also appeared that the proper functionality of the application is mostly based on the tracing of the deadlines for the event modification and registration and that similar functionality can be found in most booking systems.

The difference in this case of the application for event planning is that more than one user is concerned when a certain data is modified. Certain changes can cause the event to be cancelled that would then have an impact on all users involved in a certain event and not just on a few as in usual booking systems. That makes the event organization more customer-oriented because it must be ensured that each user was informed and not only that the data was sent to the user. To achieve that, the function, whose purpose is to regularly remind the invited persons to a certain event, was aimed to be developed as well but because of the time limitation of the project none of those two functionalities were implemented.

Other characteristics are those common to most applications where the modification of database table’s content is allowed depending on the authorization level of the user that is logged on.

5.1.1 Prerequisites and Requirements

The requirements of the application was planned to be distinguished also iteratively and this means throughout each of the phases of this project, different from the so-called “Waterfall Lifecycle” [23], where it is attempted to distinguish all requirements before the implementation phase.

To be able to run the application it is assumed that PHP 4 or a later version and MySQL are installed on the server. The AJAX application execution is depending on the browser capabilities only.

5.1.2 Software requirements specification

In the project description the MySQL and PHP are listed as non-functional requirements. The list of the software function requirements specification [24] was distinguished during the first discussion with the external client Liquidice (see “enclosure 2”).

Later on, throughout the continuous contact with the project organizer the further requirements were differentiated in each of the phases of the project.

5.1.3 Use Cases

The following use cases are described under “Enclosure 3”:

- Browse the events and event planner presentation
- Register user
- Edit user data
- Log in
• Create event
• Edit event
• Present Event
• Register Participation in Event
• Edit the database

5.2 Design

After the software requirements specification were distinguished, the most important sections of the user interface were designed, as it is presented under the enclosure 4. UML diagrams are listed under enclosure 5.

The following diagrams are presented:

• Use case diagram
• Class diagram

5.3 Implementation

During the implementation of the PHP application and MySQL database’s schemas (see xxii) and the class diagram (see xvi), made and presented previously under the subheading “Design”, were mostly used. The other diagrams were used as reminders and check-lists.

The following files are made and can be downloaded from the http://www.boro.org/EventPlanner.zip:

• index.php
• browseEvents.php
• registerUser.php
• login.php
• createEvents.php
• myAccount.php
• editEvent.php
• updateUserData.php
• registerEventParticipation.php
• connectToDatabase.php

Which requirements and use cases each php file is corresponding to can be seen from the projects “progress table” under enclosure 18.

5.4 Test

5.4.1 Verification

The system verification during this iteration was performed through its installation on two web servers and all developed functions worked as expected. That was done to ensure that the application is stable enough and that the error messages were not depending on its implementation procedure but on the differences in the PHP version installed on those servers.
5.4.2 Validation
During the first iteration the main functions were implemented, and the functions for the presentation of the created event were not considered as a priority and because of that their development was prolonged to the next iteration. It was done in that way to make more time for the testing phase.

The validation of the application under this iteration was made by Liquidice and the results are presented under enclosure 6. From the validation results the further requirements were distinguished and considered during the next iterations.

5.4.3 Screenshots
The first screenshot presents the login procedure. After having logged in, the user can edit the user data of the event he/she has authorization to edit as it is presented in the second screenshot.

The third screenshot presents the editing of the user data where all inserted data is replaced and updated.

The next screenshot presents user data after updating. A similar procedure where the first listed event is modified is done and the result can be compared with the screenshot.

6 Elaboration’s iteration 2

6.1 Analysis
During the first iteration of the elaboration phase the first prototype of the basic function was implemented and validated by the assigner of the project Liquidice. The feedback from Liquidice was used for the further development of the application.

6.1.1 Prerequisites and Requirements
To enable the possibility to upload data from some of the e-mail managers, the data must be saved as a “Comma separate values”- file (csv file) before it is uploaded to the server for mapping with the fields in the table [25].

6.1.2 Software requirements specification
The prerequisites and requirements distinguished through collaboration with Liquidice during the second iteration are described under enclosure 8.

6.1.3 Use Cases
The following use cases are presented under “Enclosure 9”:

- Store the mailing list in the database
- Read and write messages to the event’s forum
- Verify inserted data
- Browse event statistics
- Logout from the system

6.1.4 UML
The following UML diagrams are presented under “Enclosure 10”:

Activity diagram
Class Diagram
6.2 Design

The database schemas and dumpings together with the diagrams and use are presented under enclosure 20 and 21.

6.3 Implementation

The requirements distinguished after the validation in the previous iteration of the are mostly implemented. The development of some functionalities had to be modified because of the differences between the version used during the application coding and the version of the PHP on the server that is hosting the application. That had an impact on the time planned for this phase of UP.

Here are the lists of php files made during this iteration of elaboration phase that can be found on http://www.boro.org/XTevent.zip

- addressBook
- addContactList.php
- importList.php
- mapList.php
- eventForum.php
- eventStatistics.php

6.4 Test

6.4.1 Verification
The verification during this iteration of the elaboration phase was performed on three different servers and it seemed that the application executed with errors on some of its functionalities because of differences between the versions of PHP.

6.4.2 Validation
The functions implementing the requirements distinguished during the analysis, which were made within the second iteration were appropriate but some lacks in their functionality appeared during the verification of the application and are discussed later on.

The results from the validation performed by Liquidice under this iteration were used for the further development of the application during the construction phase and are presented under enclosure 11xxix

6.4.3 Screenshots
The screen shots are similar as in the next iteration with more functionalities correctly executing.
7 Construction’s - Iteration 1

7.1 Analysis

During the last iteration of the elaboration phase some lack of the application functionality appeared during the validation and had to be considered during this first iteration of the construction phase.

7.1.1 Prerequisites and Requirements

Because of the errors caused by differences between the PHP versions used for coding and the one where the application is hosted by Liquidice it was necessary to reduce the application’s functionality on the PHP version 4.3.10-19 and avoid the unnecessary object-oriented approach to the software development. That gave to the application a lot of functional programming language structure.

7.1.2 Software requirements specification

The further requirements and lacks of the application distinguished by Liquidice are listed under enclosure 11 and 12.

7.1.3 Use Cases

Use case “Add contacts” and “Edit contact” are presented under enclosure 13.

7.2 Design

The design of the interface for those two use cases appeared to be very similar to the user registration use case and because of that reason it was assumed that their implementation would be accomplished without the needs for the detailed design of their construction.

7.3 Implementation

During the first iteration of the construction phase the implementation of the remaining functionality was continued and some of the testing implementations for the next iteration use cases was also done. The implementation of the “Add contacts” and “Edit contact” use cases was done in a similar way as the user registration use case, as it was expected.

The following php files were made in this iteration of the construction phase and can be found on http://www.boro.org/XTevent.zip

- registerContact.php
- editContact.php

7.4 Test

7.4.1 Verification

During the verification of the implementation done under the first iteration of the construction phase the application was executing properly.
7.4.2 Validation
During validation under this iteration it was found that the implementation was satisfying the requirements listed by Liquidice.

The validation verified that a new design for the application layout, more compatible for the Internet Explorer browser, should be made.

7.4.3 Screenshots
On the page “myAccount.php” the user can choose to import CVS files. After that the page for the uploading and functionality for the mapping of the fields of the CVS is displayed, as the next screenshot presents. After selecting the “Edit this event” the user can also access the forum for the selected event, register for participation and view the statistics for the event participation.

8 Construction’s – Iteration 2

8.1 Analysis
During the last validation it was found that a new graphical user interface should be constructed what Liquidice accepted as appropriate suggestion.

8.1.1 Prerequisites and Requirements
During the last iteration of the elaboration phase some lacks of the application functionality appeared during the validation and had to be considered during this first iteration of the construction phase.

8.1.2 Software requirements specification
In the previous iteration of the construction phase it appeared necessary to change the design for the application to enable it to be browsable and adjusted to most of browsers. The remaining requirement “make and edit presentation for an event” from the previous iteration was left to be implemented.

8.1.3 Use Cases
The use cases “Send electronic message (E-mail)” and “Edit presentation” are presented under enclosure 15.

8.2 Design
The user interface was made by Liquidice as it was found necessary during the validation of the implementation made in the first iteration of the construction phase. This interface in form of CSS is not part of this thesis.

8.3 Implementation
The following php files was made under this last iteration of the construction phase and can be downloaded from http://www.boro.org/XTevent.zip

- sendEmail.php
- editPresentation.php
It was necessary to modify the previously developed PHP application because of the new added elements made in XHML.

8.4 Test

8.4.1 Verification

All remaining errors in the application functionality were resolved during the last iteration of the construction phase and the application was executing without the errors.

8.4.2 Validation

Liquidice validated the application and found the application sufficient. The most important functionalities appeared to be implemented with the exception of the database administration interface. The feedback from Liquidice was used to modify the visibilities of some functions.

8.4.3 Screenshots

After the user opens the first page he/she has the opportunity to use the page as an information source and if desired to register by accessing the page “Sign Up”, or if the user is already registered, to log in to the system. After having logged in, the user can access the pages “Create Event”, “My Account” and “Address Book”. After the user opens the “Create Event” page the first form for the event registration is displayed. After the user pressed the button “Save and go to step 2”, the second form is displayed. On “My Account” page the user has possibility to edit his user data, to edit selected event or to make presentation for the selected event. After the user selects to edit the presentation of a certain event he/she can create and modify the presentation for the event. The presentation is also used for the registration of the participation for the event. Because the use cases “Edit user data” and “Edit event” have similar interfaces as the “Register user” and “Register event” interfaces, it was not necessary to develop a special design for them.

When accessed the event forum, the user can view and send the message to that forum. After the user opens the “Address Book”, the opportunities for adding data for a new person into the contact list, for importing data from the new contact list and for sending e-mail to persons from the contact list are displayed.

9 Transition

The fulfilled transition of the application was not accomplished because of the project time limitation but some further feedbacks were received from Liquidice. Throughout the analysis of those feedbacks it was found that the application is implemented on a sufficient level. Most of the important functions were implemented and executing without errors while functions for confirmation of submitted data, tracking deadlines for the event modification and registration, sending invitations and reminding letters that could be done manually were not prioritized, and because the time limitation of the project was not implemented.

Presentations can be done in some advanced editor and later uploaded to the server and replace default blueprint if that does not satisfy the requirements.
Throughout the validation of the “XTevent - Event Planner”, Liquidice mainly found the functionalities, considering the database administration level not developed as expected. During the project database administration was done throughout the functions available in the php-administration tool (see http://www.phpmyadmin.net/), and because of that it was not found as a priority.

The layout, which is common for most of the pages, is required and made by Liquidice and it was found necessary to be modified. Because of that reason, place for future changes is left and the application was made able to be easily modified in contrast to the first beta version (see enclosure 7).

The further development constraints were also suggested by Liquidice and were welcomed by the project developer with the full understanding and agreement that those functions would make the present application even more interactive and user friendly than it is now.

The application developed during this project can be accessed by free registration on the page http://www.xtevent.com/registerUser.php.

An application that can be downloaded and used for event planning is also made. When a php application is used it should be considered that the layout for the page and the script codes could be added by inserting the codes into the files “headers” and “footers”.

It is possible to add whatever and that will not influence the execution of the application just its layout. Otherwise it is also possible to add code directly to each php file as well but then the code, if is not php, should be entered between php brackets. For that purpose the code is divided in php sections each starting with “<?” php and ending with “?>” signs. The combination of those two ways is recommended.
The data for database connection can be modified in the “adminlogdata.php” file.
The database tables can be created manually using the “tables dumpings” under enclosure 21 if they would not be created automatically. The “tables dumpings” can be found in index.php file also. The hardware necessary for the installation of the application is presented in the deployment diagram under enclosure 17.

10 Discussion

10.1 Open source website development

A better solution for the application development than secure one would be one where it would be allowed to create database tables. The users should be able to create at least the fields if not the tables and databases that they think are relevant and then the application should also be adapted to their modifications. That would allow them to import all data from csv files and not just those compatible to the given field names.

In this case the security was prioritized in front of flexibility, as one of the requests from Liquidice, which is not in the nature of open source. Open source should be a real evolutionary way of the software application development and that means open for the changes made by end users.
10.2 Ajax

As a client-side scripting language AJAX can be the appropriate solution for the verification of the inserted data only after the user of the system is already logged in to the system. To use AJAX before that would be reasonable only when aimed to be used e.g. for the searching of public data. Using the AJAX for searching confidential data is not recommended. It should be considered that there is the risk that event registered user could abuse the AJAX’s functionalities.

In one way it seems that the submission button has the role of a firewall that reduces the risks of the malicious data mining. It seems that because of the accessible nature of the Internet it is recommended to avoid the AJAX before the login data is verified.

What can be understood from all of this is that AJAX is suitable for the development of the interactive applications only if it is ensured that the user of the system will not abuse the access to the database.

To ensure the data security, AJAX should be used only after the user is logged in. Then can be used for data verification before data is stored into database or retrieved from it. In that way it is ensured that the access to the database is the most optimal, otherwise the database could be accessed more often than necessary, which could have an undesired influence on the server capacity.

The functions, which could be implemented in the future development of the application are those concerning the ability of the system to inform the users about the events they could be interested in to participate at. For that it would be necessary to implement the behavior tracking function which would store the information about the pages browsed by a certain user. That would require more amount of memory on the database servers but it would make the application more interactive and user friendly.

11 Conclusion

The author of this writing contributed to this project by developing php applications and an appropriate MySQL database necessary for event planning. That was done throughout the constructive analyses, design, implementation and testing during the all UP’s phases of the project development. The choice of the UP as software development process was also made by the author, which appears to be very effective for the development of this project.

Several problems occurred during the project development mostly because a new design had to be made and because of that time for detailed testing was not available. The problem could be avoided if Liquidice that insisted to make the design self, used already existed design that was tested and was working satisfactorily together with the php application. Because of these reasons some of the problems listed in the project status table (see table 1) appeared.

It would also be useful to implement the functions that would enable the application to be used for tracking the place availabilities, when the event organizer is using the same place where the events are going to occur. Then the application would track if there is a place available for a certain date and time. That would require that all bookings for those places are registered in the Event Planner database or that they are traceable for the Event Planner application in case those bookings are stored in some external database.
The cooperation with Liquidice during the project was continuous, which appeared to be in one way an unexpected disadvantage, because that caused also continuous changing of the applications user interface. It would take less time to implement the prototype and the beta version if the same person would do both the user interface and the application. The most important advantage of such continuous communication and cooperation with Liquidice was gained from the testing of the application and from the very helpful continuous feedback during the application development.

The function “Browse events” was developed but later taken away on Liquidice’s request what makes possible to browse only events the user self made. That by itself reduced the number of end users of the system to registered users only. The database administrator has access to the system throughout the php administrator tool. If there were time for the function for altering the database tables to be developed then a database administrator would not be necessary.

11.1 Project status

The final product of the application developed during this project is hosted on the http://www.xtevent.com/. The application that can be used as open source software can be downloaded from the following address: http://www.boro.org/openplanner.zip.

The project status table organized by use cases and php that lists what is and what is left to be implemented can be found under enclosure 18.

Class diagram, database schema and table dumpings are listed under enclosures 19, 20 and 21.
References


[9] Open source software, http://en.wikipedia.org/wiki/Open_source_software, (2007-03-30), ““Open source” highlights that the source code is viewable to all and proponents of the term usually emphasize the quality of the software and how this is caused by the development models which are possible and popular among free and open source software projects.”


[15] Larman, (p 33-34)

[16] Larman, p. 19

[17] Larman, p. 17-40


[23] Larman, p. 23

Enclosures

Enclosure 1

Liquidice’s announcement of “Eventplanner - Development of an event planning tool” project

Eventplanner – Development of an event planning tool

About Eventplanner

Liquidice Eventplanner is a start-up, focusing on reducing the cost of administration for company and private events by providing effective tools for the planning and execution of meeting, seminars, conferences and parties. We are now looking for creative programmers that can help us plan and develop our web based solutions for event planning. We are a part of The Rynge Group that work with transforming good ideas into great companies.

More information will shortly be available at: http://eventplanner.liquidice.com

Description of the project

We are looking for one or a few driven persons capable of writing web solutions in PHP connected to MySQL databases. You will work as a team together with the management of Eventplanner, planning and developing various web based tools for reducing the need of administration when planning events. This includes setting up a system for administrators of different levels, developing the GUI and functions for the different levels of users (eg. event organizers and event attendees) as well as interact with potential customers to get an understanding of what their demands and needs are.

Since the majority of the development will be conducted remotely, using collaborative tools, the students needs to be able to take a large portion of responsibility by themselves and the planning of the development is also a part of the work. The code and database design has to be structured and documented to make further development possible.

Preliminary schedule is set for the 10-20 week duration in the spring of 2007. However, duration and dates are flexible.

We want you to have experience in database and web solution development. We believe that the project will be rewarding for you as a person and give you the opportunity to put practical use to the knowledge you have obtained during your studies.

The project is an essential part of the success of Eventplanner, and we value your inputs and thoughts regarding the work process as well as how the solutions should be developed.

Since this is a project under development and Validation, there could be possibilities of continuance of the development if the project is successful.

To apply, please send application and resume to: ola.rynge (at) liquidice.com
**Omfattning:** 10-20 v  
**Nyckelord:** Informationshantering, Multimedia, Databaser, Databehandling, Datasäkerhet, Datavetenskap, Datagrafik, Datorkommunikation, Datorteknik, E-Handel, Gränssnitt, Informationsteknik, Internet, Nät-handel, Systemutveckling  
**Ämnesområden:** Data- och systemvetenskap  
**Geografiskt område:** Platsoberoende  
**Arbetets genomförande-period:** Vården 2007

---

**Företag/Organisation**  
**Organisationsnummer:** 790610-3994  
**Namn:** Liquidice  
**Kategori:** Tjänste-konsulter  
**Land:** Sverige  
**Antal anställda:** Färre än 50 anställda  
**Ingår I större organisation:** N

---

**Kontaktperson**  
(För företaget/organisationen)  
**Namn:** Ola Rynge  
**Postadress:** Kaptensgatan 17B  
**Postnummer:** 414 59  
**Ort:** Göteborg  
**Telefon:** 031-3609291  
**E-postadress:** ola.rynge@liquidice.com
The project is an essential part of the synergy of Competences, which enables you to apply and develop your knowledge and skills on a professional level.

Since this a project under development and evaluation, there could be possibilities of continuation of the development if the project is successful.

To apply, please send application and &v;e resume to: olv@xyga and liquid.com

On the left side of the image, there is a table and some text, but it is not clear enough to transcribe accurately.
Enclosure 2

## First iteration’s requirements

DEVELOPING PLAN

For all

- registering of user data
- feedback registration

For the administrator

Check if authentication modify/write/delete in the table user in MySQL has value “OK”

- take away or update user/event data
- delete events when outdated

For the organizer

- make event
- send invitations and specify if the invited can send them further
- specify maximum number of events
- specify if event is open only for the invited or if they can invite too
- specify the type of information (meeting, conference, travel, party)
- specify if the list of invited will be visible for all, for the invited or just for the administrator

For the invited

- specify which private info will be accessible for others
- send further invitations to other contacts if the organization allowed it
- answer yes/no/maybe
- comment the answer (specify if the answer will be visible for all/organization/invited)
Enclosure 3 : Use cases- first iteration

iii USE CASE: Browse the events and event planner presentation

1 Description: Throughout this use case the user browses the registered events presented in form of web site productions

1.1 Primary Actor: Visitor, Guest

1.2 Stakeholders and Interests
- Visitor: Expects to find enough information about the type of events he/she is interested in and eventually wants to register and to find information about the Event Planner
- Guest: Expects to find information about the event he/she is invited to and eventually will participate in
- Registered user: Expects to find enough information about the type of events he/she is interested in
- Organizer: Expects to find enough information about the events he/she has created
- Customer: Expects to find enough information about the services he/she can purchase by registration
- Company: Expects the site to be browsable
- Database administrator: Expects the events which are outdated not to be displayed if other is not requested

2 Preconditions: None

3 Deployment constraint: None

4 Flows of Events
4.1 Normal flow of events: The user browses the published events.

4.2 Alternate flow of events: The user requests more information before registration.

4.3 Exception flow of event: The user completes the form to request more information but gets informed that the Internet connection is interrupted.
**USE CASE: Register user**

1 Description: Throughout this use case the user obtains the possibility to organize the event and present it in website form for the eventual registration of the participation of the other user, as well as register his/her participation for a certain event.

1.1 Primary Actor: Guest, Customer

1.2 Stakeholders and Interests
- Guest: Expects to get information about the registration procedure
- Customer: Expects to get information about the registration procedure and confirmation about his/her eventual registration
- Company: Expects the interested visitors and guests to be able to register themselves on a user-friendly way
- Database administrator: Expects the inactive registered users to become unregistered from the database

2 Preconditions: None

3 Deployment constraint: None.

4 Flows of Events

4.1 Normal flow of events: The user registers personal data in the database.

1. The user fills in the registration form
2. The user submits the registration form
3. Gets the registration confirmation

4.2 Alternate flow of events: The user gets the message that the username he/she inserted is already in use.

1. The user submits the filled form
2. The information that the username he/she requested to be register with has already been taken is alerted
3. The user chooses an other username and submits the form
4. Gets the registration confirmation

4.3 Exception flow of event: The connection with the server is interrupted and the registration form is not submitted.

1. The user completes the registration form and submits it.
2. The user gets the alerted message displayed on the screen that the Internet connection is interrupted.
USE CASE: Edit user data

1 Description: Throughout this use case the user can change his/her registered user data

1.1 Primary Actor: Customer

1.2 Stakeholders and Interests
- Organizer: Expects his/her users data to be confidential
- Company: Expects the user to register valid user data
- Database administrator: Expects the user to keep the username and password confidential

2 Preconditions The user has to be logged in

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: The user modifies the registered information and submits the changes.

4.2 Alternate flow of events:

4.3 Exception flow of event: The user has modified the information registered for a certain event and submits the changes but gets the message that the database connection has not been established
   1. The user submits the changes
   2. The user requests the edit mode again
**USE CASE: Log in**

**1.1 Description:** Throughout this use case the user gets the opportunity to register events, edit events and register him/herself for an event.

**1.1 Primary Actor:** Registered user

**1.2 Stakeholders and Interests**
- Registered user: Expects to be able to create and edit already created events and his/hers user data
- Organizer: Expects to be able to edit already created events
- Company: Expects the login function to allow one username to be used by a single user and once at a time
- Database administrator: Expects the user not to stay logged in if inactive

**2 Preconditions:** To be able to use this user case the user has to be previously registered in the database.

**3 Deployment constraint:** None

**4 Flows of Events**

**4.1 Normal flow of events:** The user inserts the username and password and submits the inserted data.

**4.2 Alternate flow of events:** The user submits the inserted data but gets informed that the username or password is not correct.

**4.3 Exception flow of event:** The user submits the inserted data but gets informed that the Internet connection is interrupted.
USE CASE: Create event

1 Description: Throughout this use case the user registers the information about the event with the intention of the registration to the online event participation of other users.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
- Registered user: Expects to be able to create event
- Organizer: Expects created event to be browsable if requested
- Company: Expects created events to be attended and not to be unethical and against law
- Database administrator: Expects outdated events to be removed from the database

2 Preconditions: To be able to use this user case the user has to be previously registered in the database and logged in.

3 Deployment constraint: None.

4 Flow of Events
4.1 Normal flow of events: The user registers the upcoming event.
   1. The user fills in the event registration form and submits it
   2. Gets registration confirmation

4.2 Alternate flow of events: The user leaves unintentionally the event registration form without submitting it.
   1. The user partly fills in the registration form
   2. The user unintentionally leaves the browser and window registration form.
   3. The user reopens the registration form
   4. Inserted data is no longer visible in the registration form
   5. The user logs in again
   6. The user requests event registration again
   7. All or part of the previously inserted data appears in the form
   8. The user completes the event registration form and submits it
   9. Gets registration confirmation.

4.3 Exception flow of event: The user inserts unwanted data and submits the event registration form.
   1. The user fills in the event registration form and submits it
   2. The user browses the event presentation and finds unwanted data displayed
   3. The user requests event editing mode
   4. The user makes corrections and submits data
   5. Gets update confirmation.
**USE CASE: Edit event**

1 Description: Throughout this use case the user can change the information registered for the event.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
   - Organizer: Expects users he/she lists to be informed when event is updated
   - Company: Expects organizer not to update event too often to reduce the unregistration of event participation
   - Database administrator: Expects the organizer to give the authorization to edit the event only to registered users (not to share his/hers account with other persons)

2 Preconditions The user has to be logged in and authorized to edit from the user who registered the event.

3 Deployment constraint: None.

4 Flow of Events
4.1 Normal flow of events: The user modifies the information registered for a certain event and submits the changes.

4.2 Alternate flow of events: The user requests the event editing but gets informed that the event has already been modified by another user.

4.3 Exception flow of event: The user has modified the information registered for a certain event and submits the changes but gets the message that in the meantime the event has been modified.

3. The user submits the changes.
4. The changes gets saved as a draft and the user gets informed that the event he/she is trying to modify has been modified in the meantime and that those changes should be considered.
5. The user requests the edit mode again
6. The user adds the information from the draft and submits the changes
7. Gets update confirmation
USE CASE: Present Event
**************************************

1 Description: Throughout this use case the user makes presentation of the registered events.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
   • Organizer: Expects to be able to make presentation for the events he/she has created or have authorization to edit and to save the wished blueprints for the event presentations and invitations
   • Company: Expects presentations not to be unethical and against law
   • Database administrator: Expects outdated presentations to be removed from the database

2 Preconditions: To be able to use this user case the user has to be previously registered in the database and logged in.

3 Deployment constraint: None.

4 Flow of Events
4.1 Normal flow of events: The user edits the default presentation of the event.

4.2 Alternate flow of events: The user saves the edition as a draft.

4.3 Exception flow of event: The user submits the edition but gets informed that the Internet connection is interrupted.
USE CASE: Register Participation in Event

1 Description: Throughout this use case the user registers his/her participation in a certain event.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
- Visitor: Expects to be able to apply for a certain event participation
- Guest: Expects to be able to apply for a certain event participation
- Registered user: Expects to be able to register for a certain event participation
- Organizer: Expects control over the place availability of the event he/she has created
- Company: Expects registrations to certain events not to be cancelled
- Database administrator: Expects cancelled registrations to be removed from the database

2 Preconditions: The user has to be registered.

3 Deployment constraint: None.

4 Flow of Events
4.1 Normal flow of events: The user completes the participation registration and submits it.

4.2 Alternate flow of events: The user completes the registration for participation and submits it but gets informed that there is no available place for that event.

4.3 Exception flow of event: The user completes the registration for the participation and submits it but gets informed that the Internet connection has been interrupted.

   1. The user has to repeat the registration procedure
USE CASE: Edit database

1 Description: Throughout this use case the user edits the contents of the database.

1.1 Primary Actor: Database administrator

1.2 Stakeholders and Interests
   - Company: Expects accurate deletion of the data from the database
   - Database administrator: Expects automatized deletion of the data from the database

2 Preconditions: The user has to be authorized to maintain this procedure.

3 Deployment constraint: None.

4 Flow of Events
4.1 Normal flow of events: The user edits the contents of the database and submits the changes.

4.2 Alternate flow of events: The user requests to edit the database but the database is in use by some other users.

4.3 Exception flow of event: The user requests to delete a certain event but gets the message that the date of the event realization has still not passed.
Enclosure 4

xii USER INTERFACE

User registration

New User

Welcome to XTevent! You can register and join for free here.

Basic information

First Name:
Last Name:
Email Address:
Username:
Choose Password: min 6 characters
Re-enter Password:
Country: [Select Country]

Join Now

User login

Login
Username:
Password:
Remember me
Login
Event creation

Create an event - Step 1

Event Name: 
Hosted by: 
When: Apr 13 / 2007
Start Time: 20:00
End Time: 00:00 (Enter 24:00 for no end time)

Description:

Save - Go to step 2

Create an event - Step 2

Location: 
Address: 
City: 
Zipcode: 
Web: 
Phone: 
Email: 
Event Type: Private/Public
Public guest list: Yes/No
Guest invitations: No
Guest bringalongs: No
Max attendees: (optional)

Save - Go to step 3
Enclosure 5: UML diagrams - first iteration

Use case diagram
Class diagram
Enclosure 6

Validation's results -first iteration

I tested http://eventplanner.liquidice.com/EventPlanner6 (suppose that it is the latest version). Here are some comments:

- I think it looks good. I have purchased the domain xtevent.com and I think that we would be able to put the site up on that address during this week. I will work more on an updated design as well.

- If it is not too difficult, I would like the logged in box to be exchanged for a "My events"-box or something when you have logged in.

- When an event is created, I would like to be able to send invitations. The first version could be just to insert e-mail addresses manually, and then we can add an address book, and the final step is to let the user import contacts in the address book. What’s your opinion on this?

- when entering 200 as max attendees, 127 shows up under my account

- when entering 0 guest invitations / bringalongs, 8 shows up under my account

- when entering 10 guest invitations, 0 shows up under my account

- When invitations are sent, we need an interface for accepting responses as well.

Please provide feedback on these comments.
Enclosure 7: Screenshots- first iteration
Enclosure 8

Second iteration requirements - the elaboration phase

- The mailing list in form of the comma separated value files .csv is possible to be stored in the database
- The reminder function for sending letters to the invited guests when the number of days to the event is less than the given to remind them of the event or to register if they still did not
- Contacts list categorization
- Function which gives opportunity to those from the contact list to self use the list
- Save the information about the e-mails which failed to be sent
- Function which gives alternatives for the treatment of those e-mails (date for resending)
- Event forum
- Event statistics
- Inserted data verification
- Logout function
USE CASE: Store the mailing list in the database

1 Description: Throughout this use case the user stores wanted mailing list into the database

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests

- Registered user: Expects to be able to choose the e-mail from the list and add new ones
- Organizer: Expects to be able to choose the e-mail from the list and add new ones
- Company: Expects the list to be editable
- Database administrator: Expects the list to be editable

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events

4.1 Normal flow of events:

User selects the file containing the list wished to be stored.
User stores the list into the database. Chooses the e-mail address from the list and sends a message to that address.

4.2 Alternate flow of events:

Choses the date for the reminder e-mails to be sent.
Categorizes the mailing list.

4.3 Exception flow of event:

User deletes unintentionally the e-mail from the list.
User chooses the e-mail from the list and sends the e-mail that failed to be sent.
User gives the alternatives for the treatment of the messages that failed to be sent.
USE CASE: Read and write messages to the event’s forum

1 Description: Throughout this use case the user is able to read and add messages to the forum for the certain event

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
   - Registered user: Expects to be able to read and write messages to the forum
   - Organizer: Expects to be able to read and write messages to the forum
   - Company: Expects the forum to be accessible for all visitors
   - Database administrator: Expects the forums for the old expired events to be removed from the event after the given date

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: User fills in the form and presses the submit button

4.2 Alternate flow of events:

4.3 Exception flow of event: User fills the form but closes unintentionally the forum page. User has to fill in the form again.
**Use Case: Verify inserted data**

1. **Description:** Throughout this use case the user gets informed if all required data is not inserted.

1.1 **Primary Actor:** Organizer

1.2 **Stakeholders and Interests**
- Visitor: Expects to get the message when all required data is not inserted
- Guest: Expects to get the message when all required data is not inserted
- Registered user: Expects to get the message when all required data is not inserted
- Organizer: Expects to get the message when all required data is not inserted
- Company: Expects the users to get the message when all required data is not inserted

2. **Preconditions:** None

3. **Deployment constraint:** None

4. **Flow of Events**

4.1 **Normal flow of events:**
User does not fill in all data and presses the submit button.
User is asked to fill in those remaining fields.

4.2 **Alternate flow of events:** None

4.3 **Exception flow of event:**
User fills the form but closes unintentionally the forum page.
User has to fill in the form again.
*** USE CASE: Browse event statistics

1 Description: Throughout this use case the user gets the participation statistics about a certain event.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
  - Registered user: Expects to get information about the number of participants
  - Organizer: Expects to get information about the number of participants
  - Company: Expects the users to get information about the number of participants

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: User opens the page where the number of users are presented

4.2 Alternate flow of events: None
4.3 Exception flow of event: None
**USE CASE: Logout from the system**

1 Description: Throughout this use case the user logs out and on this way ensures that nobody who has no authorization can view and modify data.

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
   - Registered user: Expects the data not to be visible after he/she logs out
   - Organizer: Expects the data not to be visible after he/she logs out
   - Company: Expects the user to be able to log out

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
   4.1 Normal flow of events: User presses the logout button

   4.2 Alternate flow of events: None

   4.3 Exception flow of event: None
Enclosure 10: UML activity and class diagram

Activity Diagram

Class Diagram
Validation's results - second iteration

Here are some comments on the functions:

Contacts:
• When mapped the fields, no button to finalize the import.

• Should be a field for mapping called "not mapped" or something for the fields you don’t want to include in the address book.

• How do I add contacts manually?

• How do I show my contacts?

The event pages:
• If I include a " in the description it turns out as \\

• Still 127 is the maximum attendance that can be input

• The Zipcode doesn’t register correctly either

• If I register, it stills shows 0 persons attending the event.

• I can register even if I am not invited...

Invitations:
I still can't find a way to invite people to the events

Registrations:
When a invitee replies, they should be able to select Yes, No or Maybe and add a comment.
Enclosure 12

First iteration’s requirements – Construction phase

- add contacts manually
- show my contacts
- edit contacts
- make and edit presentation for en event
Enclosure 13: Construction phase: Use cases - the first iteration

**********************************************************

xxx USE CASE: Add contact
**********************************************************

1 Description: Throughout this use case the user adds the information about a certain person to the contact list

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
   • Registered user: Expects to be able to add a contact and view it
   • Organizer: Expects to be able to add a contact and view it
   • Company: Expects to be able to add a contact and view it

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: User inserts information about a certain person and presses the submit button

4.2 Alternate flow of events: None

4.3 Exception flow of event: User inserts information about a certain person and presses submit but gets informed that the required fields are not inserted
USE CASE: Edit contact

1 Description: Throughout this use case the user edit the information about the certain person from the contact list

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests
- Registered user: Expects to be able to edit contact and view it
- Organizer: Expects to be able to edit contact and view it
- Company: Expects to be able to edit contact and view it

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: User edit information about the certain person and press submit button

4.2 Alternate flow of events: None

4.3 Exception flow of event: User finish editing the information about the certain person and press submit but gets informed that the required field are not inserted
Enclosure 14: Screenshots- first iteration

xxxiii
Register Yourself For The Event
Welcome to XTevent! You can register for free here.

Basic Information
First Name:
Last Name:
Email Address:

Sign up
Signing up allows you to create events and keep all of your event data in one place. Please fill in the details on the registration page to create a login for you.

Login
Username:
Password:
Login

Logout
STATISTICS

THERE ARE 2 PERSONS PARTICIPATING TO THE SELECTED EVENT!
Enclosure 15: Use cases- the second iteration

******************************************************************************

xxxix USE CASE: Send electronic message (E-mail)
******************************************************************************

1 Description: Throughout this use case the user is able to read and add messages to the forum for a certain event

1.1 Primary Actor: Organizer

1.2 Stakeholders and Interests

- Registered user: Expects to be able to send e-mails
- Organizer: Expects to be able to send invitations throughout the e-mail function and that it is ensured that the e-mail has been sent

2 Preconditions: None

3 Deployment constraint: None

4 Flow of Events
4.1 Normal flow of events: The user fills in the e-mail form and presses the submit button

4.2 Alternate flow of events: The user fills in the e-mail form and presses the submit button and gets informed that all fields are not inserted

4.4 Exception flow of event:
The user fills the form but the e-mail is not sent
The function for the tracking of unsent e-mails ensures that the e-mail will be sent
**USE CASE: Edit presentation**

1. **Description:** Throughout this use case the user is able to make and edit a presentation for a certain event

1.1 **Primary Actor:** Organizer

1.2 **Stakeholders and Interests**

- Registered user: Expects to be able to view a presentation
- Organizer: Expects to be able to edit the presentations
- Company: Expects the presentations to be accessible for all visitors
- Database administrator: Expects the old presentations for the expired events to be removed from the event after the given date

2. **Preconditions:** None

3. **Deployment constraint:** None

4. **Flow of Events**

4.1 **Normal flow of events:** The user sets the presentation properties and presses the submit button

4.2 **Alternate flow of events:** The user sets the presentation properties and presses the submit button but is not satisfied with its layout
   - The user repeats the procedure

4.3 **Exception flow of event:** The user sets the presentation properties and presses the submit button but the settings are not registered
   - The user has to repeat the procedure
Enclosure 16: Screenshots - second iteration

XTevent - event & meeting planning made easy

XTevent offers tools for planning and execution of events ranging from large company events such as conferences, seminars and meetings to private events like parties or vacation trips between friends.

The purpose of XTevent’s tools is to reduce the cost and time spent on planning and organizing events. Example of time consuming tasks when organizing events are: administrating invitations, booking accommodations, searching and booking suitable event facilities and administrate invoices and payments.

By using XTevent for the organization of events, the organizer will have access to the current event information at all time, gathered at the same place.

Beta testing open

Currently we are looking for customers (in south and west Sweden) to beta test and develop the system together with us. Send an e-mail to beta@xtevent.com for more info.

Work for XTevent

We are always looking for talented people. Look under jobs to view our current openings.
Welcome Borislav

To the right you can see the status of your current events. Click the event to get more detailed information.

Current Events

2007-06-07
Frukostmöte Connect Väst
Fully booked (350 attendees)

2007-06-11
Premiär Citron & Timjan
Yes: 120 Maybe: 32 No: 250

2007-06-14
Frukostmöte Connect Väst
Invitations not sent yet

> Go to event history
My Contact List

Upload the contact list you want to be stored into the database!

Choose File  no file selected
UPLOAD THE FILE

FirstName
LastName
Email
Website
IMPORT THE CONTACT LIST

Current Events
2007-06-07
Frukostmøte Connect Väst
Fully booked (350 attendees)

2007-06-11
Premiär Citron & Timjan
Yes: 120 Maybe: 32 No: 250

2007-06-14
Frukostmøte Connect Väst
Invitations not sent yet

> Go to event history
Deployment diagram

Enclosure 17
### Project status table

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Use case</th>
<th>Php-script</th>
<th>Implemented</th>
<th>Function</th>
<th>Function with errors</th>
<th>Not implemented</th>
</tr>
</thead>
<tbody>
<tr>
<td>- browse registered events</td>
<td>Browse the events and event planner presentation</td>
<td>- index&lt;br&gt;- browseEvents</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Liquidice has decided not to allow this function before the user is logged in</td>
</tr>
<tr>
<td>- registering user data</td>
<td>Register user</td>
<td>- registerUser</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- take away or update user data</td>
<td>Edit user data</td>
<td>- updateUserData</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- login to the system</td>
<td>Log in</td>
<td>- login</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- make event</td>
<td>Create event</td>
<td>- createEvent</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>function &quot;give authorisation to the other user to edit the event&quot; is not implemented</td>
</tr>
<tr>
<td>- specify if the invited can send them further</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- specify maximum number of events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- specify if the event is open only for the invited or if they can invite too</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- specify the type of information (meeting, conference, travel, party)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- specify if the list of invited will be visible for all, for the invited or just for the administrator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- give authorisation to the other user to edit the event</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- take away or update event data</td>
<td>Edit event</td>
<td>- editEvent</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Requirements</th>
<th>Use case</th>
<th>Php-script</th>
<th>Implemented</th>
<th>Function</th>
<th>Function with errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>- make presentation for the events he/she has created or have authorization to edit</td>
<td>Present Event</td>
<td></td>
<td>No</td>
<td></td>
<td>Satisfies the requirement “make presentation for the events he/she has created or have authorization to edit” throughout the “editEvent.php”</td>
</tr>
<tr>
<td>- save the wished blueprints for the event presentations and invitations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- answer yes/no/maybe</td>
<td>Register Participation in Event</td>
<td>-registerEventParticipation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- check place availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- comment the answer (specify if the answer will be visible for all/organization/invited)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- specify which private info will be accessible for others</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- send further invitations to other contacts if the organization allowed it</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- check if guests can invite and/or bring along persons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- send automatically the reminding letter to event participants</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- automatically delete events and presentations when outdated</td>
<td>Edit the database</td>
<td></td>
<td>No</td>
<td></td>
<td>Because of the time limitation first function is let to be done or manually by the users</td>
</tr>
<tr>
<td>- specify maximum number of events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- event statistics</td>
<td>Browse event statistics</td>
<td>- eventStatistics.php</td>
<td>Yes</td>
<td>It displays only event participation statistics</td>
<td>Function include those who answered “no” and “maybe” to a number of participators</td>
</tr>
<tr>
<td>- event forum</td>
<td>Read and write messages to the event’s forum</td>
<td>- eventForum</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- feedback registration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- the mailing list in form of comma separated value files .csv possible to be stored in the database</td>
<td>Store the mailing list in the database</td>
<td>- addContactList</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- contact list categorization</td>
<td></td>
<td>- importList</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- mapList</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>Use case</td>
<td>Php-script</td>
<td>Implemented</td>
<td>Function</td>
<td>Function with errors</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
<td>-------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>- inserted data verification</td>
<td>Verify inserted data</td>
<td>- registerUser</td>
<td>Yes</td>
<td>This function is</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- createEvent</td>
<td>made with JavaScript</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- logout function</td>
<td>Logout from the system</td>
<td>- login</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>- add contacts manually</td>
<td>Add contacts</td>
<td>- registerContact</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>- contact list categorization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- show my contacts from the contact list by its category</td>
<td>Edit contact</td>
<td>- editContact</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Function for searching the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>contacts is made both in</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>AJAX and php</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- send invitations</td>
<td>Send electronic message</td>
<td>- sendEmail</td>
<td>Yes</td>
<td>Satisfy requirements:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(E-mail)</td>
<td></td>
<td></td>
<td>&quot;- send invitations&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>and</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&quot;save the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>information about</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>the e-mails which</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>failed to be sent</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Resending</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>messages is not</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>properly tested</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make and edit the presentation for an event</td>
<td>Edit presentation</td>
<td>- editPresentation</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>All attributes for headings are not implemented</td>
</tr>
</tbody>
</table>
Class diagram describing relations between the tables from database schemas
Enclosure 20

### Database schemas

#### Table: User

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>int(11)</td>
<td>Primary</td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>tinytext</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table: Edits

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>int(11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventNr</td>
<td>int(11)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table: Event

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventNr</td>
<td>int(11)</td>
<td>Primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserID</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventName</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VisibilityStatus</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegistrationDate</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Price</td>
<td>float</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info</td>
<td>text</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CreatedBy</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegistrationDeadline</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventEditDeadline</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartDate</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndDate</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartHour</td>
<td>tinyint(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartMinute</td>
<td>tinyint(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndHour</td>
<td>tinyint(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndMinute</td>
<td>tinyint(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>text</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zipcode</td>
<td>smallint(6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WwwAddress</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventType</td>
<td>tinytext</td>
<td></td>
<td>Business, Fun, Travel, Seminars</td>
<td></td>
</tr>
<tr>
<td>GuestsAllowed</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuestsInvitation Allowed</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GuestBringalongs</td>
<td>tinyint(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxAttendees</td>
<td>tinyint(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table: Contacts

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListNr</td>
<td>int(11)</td>
<td>Primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserID</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>int(11)</td>
<td></td>
<td>Business, Fun, Travel, Seminar</td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastName</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BusinessPhone</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HomePhone</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fax</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MobilePhone</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompanyName</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Website</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>City</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZipCode</td>
<td>tinyint(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendEmail</td>
<td>tinytext</td>
<td></td>
<td>Yes; No</td>
<td></td>
</tr>
<tr>
<td>Country</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table: Forum

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventNr</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>date</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table: Participates

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParticipationNr</td>
<td>int(11)</td>
<td>Primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventNr</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FirstName</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lastname</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Answer</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coments</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BringWith</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table: Presentation

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PresentationNr</td>
<td>int(11)</td>
<td>Primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EventNr</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LastModified</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UserID</td>
<td>int(11)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logo</td>
<td>mediumtext</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table: Heading

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeadingNr</td>
<td>tinyint(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PresentationVersion</td>
<td>bigint(20)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontType</td>
<td>char(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontSize</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontColor</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text</td>
<td>mediumtext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontLink</td>
<td>mediumtext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontEmail</td>
<td>mediumtext</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: SendEmails

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Key</th>
<th>Valid values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td>text</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SendEmail</td>
<td>tinytext</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enclosure 21

*vi* Table dumpings

```sql
-- Table structure for table `User`

CREATE TABLE `User` (
  `UserID` int(11) NOT NULL auto_increment,
  `FirstName` tinytext NOT NULL,
  `LastName` tinytext NOT NULL,
  `UserName` tinytext NOT NULL,
  `Password` tinytext NOT NULL,
  `Email` tinytext NOT NULL,
  `Country` tinytext NOT NULL,
  `BrowserID` tinytext NOT NULL,
  `SessionID` tinytext NOT NULL,
  PRIMARY KEY  (`UserID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=17 ;

-- Dumping data for table `User`

--

-- Table structure for table `Edits`

CREATE TABLE `Edits` (
  `UserID` int(11) NOT NULL,
  `EventNr` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Dumping data for table `Edits`

--
```
-- Table structure for table `Event`

CREATE TABLE `Event` (
    `EventNr` int(11) NOT NULL auto_increment,
    `UserID` int(11) NOT NULL,
    `EventName` tinytext NOT NULL,
    `VisibilityStatus` tinyint(1) NOT NULL,
    `RegistrationDate` date NOT NULL,
    `Price` float NOT NULL,
    `Info` text NOT NULL,
    `CreatedBy` tinytext NOT NULL,
    `RegistrationDeadline` date NOT NULL,
    `EventDateEditDeadline` date NOT NULL,
    `StartDate` date NOT NULL,
    `EndDate` date NOT NULL,
    `StartHour` tinyint(4) NOT NULL,
    `StartMinute` tinyint(4) NOT NULL,
    `EndHour` tinyint(4) NOT NULL,
    `EndMinute` tinyint(4) NOT NULL,
    `Description` text NOT NULL,
    `Location` tinytext NOT NULL,
    `Address` tinytext NOT NULL,
    `City` tinytext NOT NULL,
    `WwwAddress` tinytext NOT NULL,
    `Phone` tinytext NOT NULL,
    `Email` tinytext NOT NULL,
    `EventType` tinytext NOT NULL,
    `GuestsAllowed` tinytext NOT NULL,
    `GuestsInvitationAllowed` tinyint(1) NOT NULL,
    `GuestBringalongs` tinyint(1) NOT NULL,
    `MaxAttendees` int(1) NOT NULL,
    PRIMARY KEY  (`EventNr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=212 ;

-- Dumping data for table `Event`

-- Table structure for table `Contacts`

CREATE TABLE `Contacts` (
    `ListNr` int(11) NOT NULL auto_increment,
    `UserID` int(11) NOT NULL,
    `Cathegory` tinytext NOT NULL,
    `FirstName` tinytext NOT NULL,
    `LastName` tinytext NOT NULL,
'Email' tinytext NOT NULL,
'BusinessPhone' tinytext NOT NULL,
'HomePhone' tinytext NOT NULL,
'Fax' tinytext NOT NULL,
'MobilePhone' tinytext NOT NULL,
'CompanyName' tinytext NOT NULL,
'Website' tinytext NOT NULL,
'Address' tinytext NOT NULL,
'City' tinytext NOT NULL,
'ZipCode' tinyint(5) NOT NULL,
'SendEmail' tinytext NOT NULL,
'Country' tinytext NOT NULL,
PRIMARY KEY ('ListNr')
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

--
-- Dumping data for table `Contacts`

-- --------------------------------------------------------

-- Table structure for table `Forum`
--

CREATE TABLE `Forum` (
  `EventNr` int(11) NOT NULL,
  `Title` tinytext NOT NULL,
  `Message` text NOT NULL,
  `Author` tinytext NOT NULL,
  `Date` date NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `Forum`
--

-- --------------------------------------------------------

-- Table structure for table `Participates`
--

CREATE TABLE `Participates` (
  `ParticipationNr` int(11) NOT NULL auto_increment,
  `EventNr` int(11) NOT NULL,
  `FirstName` tinytext NOT NULL,
  `LastName` tinytext NOT NULL,
  `Email` tinytext NOT NULL,
  `Answer` tinytext NOT NULL,
  `Comments` text NOT NULL,
  `BringWith` tinytext NOT NULL,
  PRIMARY KEY (`ParticipationNr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=55 ;

--
-- Dumping data for table `Participates`

85
-- Table structure for table `Presentation`

CREATE TABLE `Presentation` (  
  `PresentationNr` int(11) NOT NULL auto_increment,  
  `EventNr` int(11) NOT NULL,  
  `LastModified` int(11) NOT NULL,  
  `UserID` int(11) NOT NULL,  
  `Logo` tinytext NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=274 ;

-- Dumping data for table `Presentation`

-- Table structure for table `Heading`

CREATE TABLE `Heading` (  
  `HeadingNr` tinyint(4) NOT NULL,  
  `PresentationVersion` bigint(20) NOT NULL,  
  `FontType` char(1) NOT NULL,  
  `FontSize` tinytext NOT NULL,  
  `FontColor` tinytext NOT NULL,  
  `TextPosition` tinytext NOT NULL,  
  `Text` text NOT NULL,  
  `FontLink` mediumtext NOT NULL,  
  `FontEmail` mediumtext NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Dumping data for table `Heading`

-- Table structure for table `SendEmails`

CREATE TABLE `SendEmails` (  
  `Email` tinytext NOT NULL,  
  `Title` tinytext NOT NULL,  
  `Author` tinytext NOT NULL,  
  `Message` text NOT NULL,  
  `SendEmail` tinytext NOT NULL,  
  `Time` int(12) NOT NULL  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Dumping data for table `SendEmails`