

MASARYK UNIVERSITY
FACULTY OF INFORMATICS



Information System of the Crisis and Intervention Centre

BACHELOR THESIS

Radoslav Husár

Brno, May 2009

Declaration

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Advisor: prof. Ing. Václav Přenosil, CSc.

Acknowledgement

I am very grateful to my advisor, prof. Ing. Václav Přenosil CSc., for his advise and prompt support with all issues that occurred on the way as well as giving me the opportunity to work on an interesting Bachelor thesis.

I would like to thank Jiří Hrabálek and his colleagues at Spondea for demonstrating the current system and specifying needs for the new information system. Thanks to Bc. Tomáš Pavelka who gathered all required fields and whose database table structure from his Bachelor thesis was synchronized with model used in the prototype.

A special thanks is for my friends and colleagues for long discussions about Java and open source projects and looking for the best solutions to all kinds of design and implementation challenges.

Abstract

The aim of this thesis is to analyze, design and develop a prototype of an information system for a crisis and intervention centre. It strives to show the advantages of leveraging the latest cutting edge technology in Java Enterprise Edition architecture which is based on open standards and specifications. Object-oriented analysis is done using Unified Modeling Language, implementation is based around Seam framework taking advantage of the bundled tools.

Keywords

information system, object-oriented analysis and design, UML 2, Java, Java Enterprise Edition 5, Seam framework, MySQL, JBoss, Hibernate

Contents

1	Introduction	1
2	Analysis of current state	3
2.1	<i>Workflow</i>	3
2.2	<i>Problems</i>	4
2.2.1	Redundancy	4
2.2.2	Concurrency	4
2.2.3	Authorization	4
2.2.4	Multiple sources of data	4
2.2.5	Search	5
2.2.6	Statistics	5
2.2.7	Agreement printing	5
2.3	<i>Hardware</i>	5
3	Analysis and design of the information system	6
3.1	<i>Textual analysis</i>	6
3.2	<i>Use case diagram</i>	7
3.3	<i>Class diagram</i>	8
3.3.1	Analysis	8
3.3.2	Design	9
3.4	<i>Deployment diagram</i>	10
4	Technology selection	12
4.1	<i>Criteria</i>	12
4.1.1	Free and open sourced	12
4.1.2	Platform independent	13
4.1.3	Robust, scalable and reliable	13
4.1.4	Supported	14
4.1.5	Secure	14
4.2	<i>Java Enterprise Edition 5</i>	14
4.3	<i>Seam framework</i>	15
4.4	<i>JavaServer Faces, Facelets</i>	17
4.5	<i>RichFaces</i>	18
4.6	<i>JBoss Application Server 5</i>	18
4.7	<i>MySQL database</i>	19
4.8	<i>Hibernate</i>	20
4.9	<i>Development environment</i>	21
5	Implementing operational information system	22
5.1	<i>Preparing SQL</i>	22
5.2	<i>Using seam-gen</i>	24
5.3	<i>Bottom-up and top-down</i>	25
5.4	<i>Development with Seam</i>	27
5.5	<i>Seam components</i>	27

5.5.1	Home object	28
5.5.2	Query object	29
6	Conclusion	31
	Bibliography	32
	Index	33
A	Contents of the attached CD	34
B	Deployment diagram	35

Chapter 1

Introduction

Spondea is generally beneficial organization whose purpose is to improve quality of life of children, young people with their relatives and people suffering or threatened by home violence. The aim is to help them improve their abilities to deal with their complicated life situation and successfully incorporate into society again.

Since its establishment in November 1998, Spondea helped thousands of young people in need. The range of the services provided and the temporary accommodation capacity makes it the only one organization of such type in the entire area of southern Moravia. The offer of social services can be summarized into four categories. Nonstop personal crisis help, nonstop crisis help over the telephone, social service for families with children and the intervention center open nonstop as well.

However, as other non-profit organizations, it is required to keep record of all activities which namely are clients and their contacts with the social workers, as well as provided temporary accommodation. This information is later used to generate statistics and reports for the donors who support the organization financially. Lack of software for this purpose then results in a lot of work to be done by the social workers who could be spending more time helping children.

My motivation was to work on software that eventually could be deployed and used and that my work will be helpful to someone. Other non-profit organizations might be in a similar need so it can become a community maintained open source project and possibly developed further. This Bachelor thesis gave me a perfect opportunity to develop an information system that can be used – Spondea o.p.s. based in Brno – and know that my work is helping real people.

Moreover, I want to use a Bachelor thesis as an opportunity to explore and demonstrate new technology. I decided to search for the best and latest technology that is available today. I want to harness the power of the latest cutting-edge technology to prove that not only that this technology is usable, but also that it is user-friendly to develop a robust and secure application that can be later deployed and reliably managed. Object-oriented analysis and design was done, models were created using Unified Modeling Language. Software selection criteria was based on reliable and robust yet standards-compliant and free software.

This thesis is divided into four chapters. First chapter explains the current state of information technology and the workflow, pinpointing its main problems. Second chapter exploiting the previous one, analysis the requirements of an information system that could replace currently used software in the organization. As a result a new design is suggested.

Third chapter deals with establishing criteria for selecting technology to be used and afterwards presents the selected software, highlighting its benefits. Fourth chapter explains the software development patterns that has been used for development of the information system, showing main principles and advantages opposed to older patterns.

Chapter 2

Analysis of current state

In this chapter I will describe the current workflow in this organization and problems it causes due to a lack of proper software infrastructure. I have spent time with infrastructure administrator and the social workers at Spondea o.p.s. in Brno discussing the current workflow, analyzing the problems and redundant work they have to face.

2.1 Workflow

Spondea o.p.s runs three different centres:

- Crisis centre (Krizové centrum, KC) is helping children and youth,
- Intervention centre (Intervenční centrum, IC) which is part of the Donacatrum, is helping victims of home violence when it resolved to the banishment of a family member,
- Donacetrum (Donacentrum, DC) is helping people who are threatened by home violence, but in which case a family member was not banished.

Spondea offers help to people non-stop. To maximize the availability, Spondea offers a big variety of different communication channels. Each contact with the centre has to be reported in the daily report. The term “contact” refers to every interaction through any of the communication channels the centre provides at the moment – telephone, personal, SMS, e-mail, etc. Currently, this is being recorded in a Microsoft Excel sheet shared over the network.

In case the person stays for a longer time, which mainly includes personal (ambulant) contact, this has to be logged in the daily report as well with a brief description. The detailed description of the session is recorded in the database for interventions by the doctor. Currently, this is a custom simple application built on Microsoft Access.

Spondea also offers temporary accommodation for people who need it. In that case, client’s daily routine has to be recorded in the daily routines document. An agreement has to be printed and signed with the client. A template is edited by the employee and printed out to be signed by the client.

At the end of each month, the national law requires the organization to produce a report. Data has to be carefully selected for proper categories, gathered together and statistics are then generated by the administrator.

2.2 Problems

This section summarizes problems that the organization's current workflow faces and that can or cannot be improved, trying to provide a solution to eliminate or avoid them.

2.2.1 Redundancy

As shown in the previous section, the workflow in the company is not very complicated. The problematic part is that every activity has to be recorded, because it is required by the legislation. Very often the clients of one centre become or already are clients of the other 2 centres, though data have to be separately managed and organized for the crisis centre and for the intervention centre. This leads to multiple duplicates and it would be helpful if the data about clients could be shared between centres. However this is not allowed by the legislation and therefore 2 different and independent information systems have to be operating in this organization as a whole. Employees have different access to the data, some can access data only of the crisis centre and some of them only intervention centre data.

2.2.2 Concurrency

Initial problem is that each contact with the centre has to be reported in the daily report. Contact is every interaction through any communication channel the centre provides – telephone, personal, SMS, e-mail, etc. Since there is no central information system that could manage a concurrent data access, the centre uses a simple Microsoft Excel file shared over the local network. However, this causes many problems with concurrent users trying to work on the same document. Microsoft Excel at the used version locks the file on the file system and no other user can edit this document. It is an understandable behaviour, because it was not designed for such purpose.

2.2.3 Authorization

Authorization is therefore handled on the computer log in level, which connects to the Microsoft domain and is granted access rights to the file. Even though the information system will be only available through on the local network, user authorization should be handled on the future information system.

2.2.4 Multiple sources of data

The data about interventions are stored in Microsoft Access. It uses a database which opens on single requests, which behaves like transactions, but since the load is really low, the possible performance degradation does not represent a problem. Data are however scattered around in different files. Information system should provide unified access to all necessary data as well as a single database can provide unified data access to the administrator.

2.2.5 Search

Microsoft Excel or the current spreadsheet design also does not offer features that information system can provide to improve user experience and decrease time taken working with the system. Information system can provide advances search criteria with and quick sorting, options to merge two clients who are found to be the same person, data integrity, associations, browsing, updates, etc.

2.2.6 Statistics

Towards the end of each month, employees working with the clients go through each client contact and copy it to another database which is then used by the administrator to generate statistics. The immediate problem is having the same data in two locations since they can be marked in the system whether to count it or not. However, generating global statistics in the information system would be a different problem level. This is because the legislation is changing frequently. The system would need to have an expression evaluator which would gather selected records and calculate given statistical data on them. The agreement has been reached that the statistical data will be handled outside of the system by the administrator who can even be generating graphs and still be able to customize their appearance in an application designed for such purpose. The exchange output file will be Microsoft Excel sheet.

2.2.7 Agreement printing

When a client is accommodated an agreement has to be printed out and signed by the client or his representative. Since the data which is printed are the same as in the information system, it should provide document generation for printing and eliminate copying the data from one document to another.

2.3 Hardware

As a part of the analysis of the current state and for future reference on what can the organization afford to deploy, I was shown the current hardware infrastructure at Spondeo o.p.s. in Brno.

All computers in the organization are running Microsoft Windows XP, connecting to one domain server. All computers are connected with fast wired Ethernet network and are also connected with the main server and other servers, therefore an intranet application is an ideal solution. Data has to be backed up on daily basis. Unfortunately there is no geographical backup, only locational on different floors in the building. System should be ready with backup tools to enable the administrator to schedule and manage automatic backup effortlessly.

Additional hardware may be purchased from the donations requested for a particular cause if the current servers cannot be used due to performance or reliability issues.

Chapter 3

Analysis and design of the information system

In this chapter object-oriented analysis will be done, starting with textual analysis to analyze the problem domain to the class and deployment diagrams. As this Bachelor thesis strives to be a manifestation of modern object-oriented technology, diagrams are modelled using Unified Modeling Language (UML). UML is a standardized general-purpose modeling language for software engineering. Used UML 2.0 major revision was standardized by Object Management Group (OMG) in 2005 including a set of graphical notation to model abstract situations of specific systems.

3.1 Textual analysis

Textual analysis is the basic document of requirement capturing. It is a process to analyze the system domain. It helps identifying the candidate classes in a problem statement to provide a better view of the problem for further analysis.

Textual analysis itself can also serve well for communication with the customer, since understanding it does not require any specialized knowledge unlike UML diagrams in next sections. An example can be seen in Figure 3.1. [9]

Information system of crisis center, as a part of Spondeo o.p.s., manages clients, contacts, stays and employees. Employees can create a new client with a contact and be duplicated in case of multiple clients coming together. Existing clients may be browsed and searched for according to criteria. All information about the client is optional. For an existing client a new contact can be created. Clients can be merged into one client as more information becomes available and duplicity is detected. Contacts and clients can be browsed and edited. Clients can be deleted when they have no contacts and stays. When a client is accommodated an agreement is signed with the client. Then for each day a day log can be appended. System administrator can export data to create reports and statistics and manage employees.

Figure 3.1: Textual analysis

Following Figure 3.2. is the textual analysis of the problem, extracted keyword for candidate classes and actors, operations for use cases opened in Visual Paradigm software.

Textual analysis provides an overview of the candidate classes, use cases and actors.

3.2. USE CASE DIAGRAM

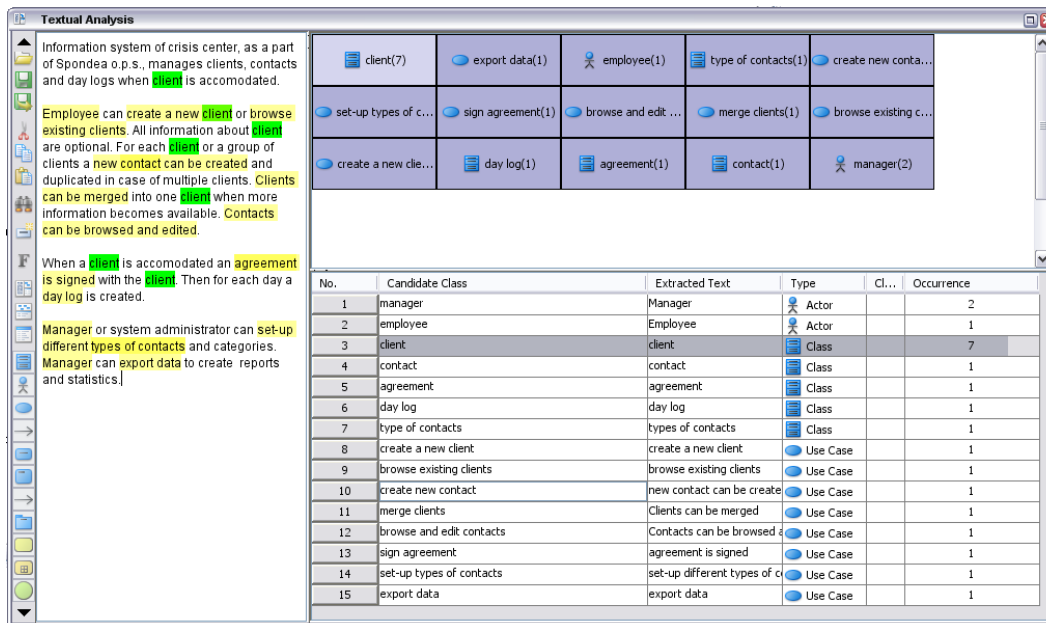


Figure 3.2: Textual analysis in Visual Paradigm

There are 2 important roles – actors – in the system – *employee* and *system administrator*. The candidate classes, which will probably become the persisted objects in the database, are *client*, *contact*, *agreement* and *employee*. Basic use cases are shown and discussed in the next section regarding use case modeling in UML.

3.2 Use case diagram

Use case diagram in the UML is a type of behavioural diagram created as an outcome of use case analysis process. Its main purpose is to show a functionality provided by a system to actors in form of use cases, goals users want to achieve, and possible dependencies between use cases in a graphical overview. Diagram shows the context of one use case in regard to other use cases. As a communication artifact, a group of comprehensible use cases provides a useful picture of system and its expected behavior to help understanding between the customer and the development team.

The notations of a use case diagram in UML is simple to enable understanding of the diagram by the non-technical customer as well. Use cases are represented by ovals with the name of the activity in it and the actors are represented by stick figures with their role names below the figure. The association between them is denoted by a solid line. System is bounded by the rectangle and use cases are placed in it. Generalization and specialization is denoted by the general actor, which can perform common actions, having arrows pointing towards it from the specialized actor. Use cases can include or provide extension points by

dotted arrows. [8]

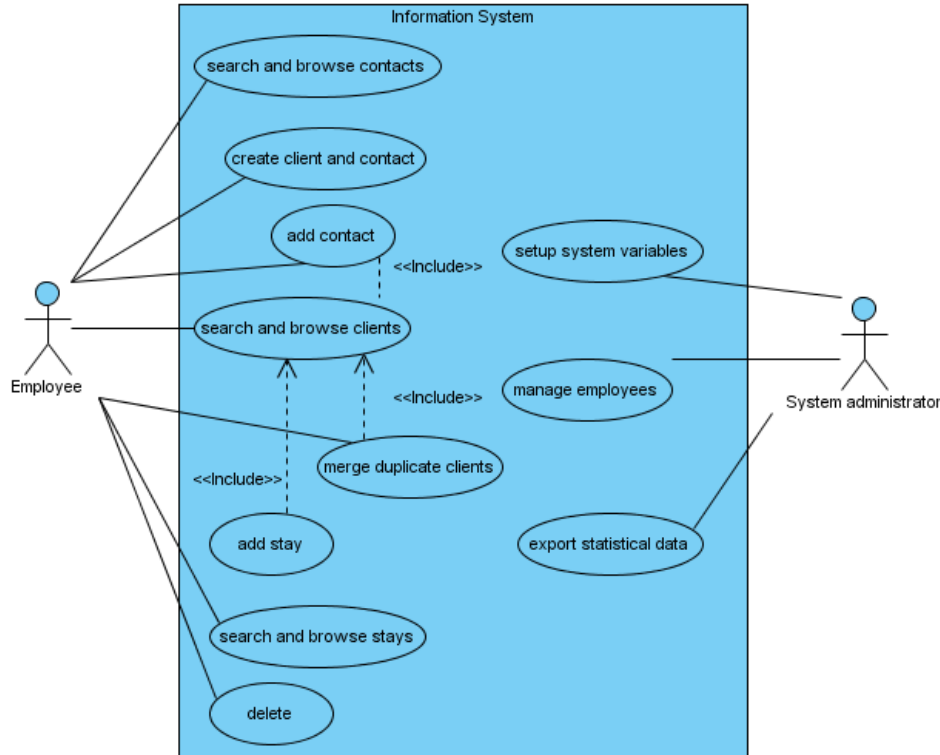


Figure 3.3: Use case diagram

3.3 Class diagram

Class diagrams in UML 2 specification are the the core of object-oriented analysis and design. They represent static structures – classes of the system, depict relationships, attributes of the classes and supported operations on them. Variety of relationships shows what kind of relationship is between two classes including aggregation and association, inheritance, generalization and specialization or simple usage. Class diagrams can be used in a different extends for different purposes. Primarily for conceptual problem domain modeling and for deeper design modeling. Both diagrams will be shown. [8]

3.3.1 Analysis

The basic classes, without including the classes that provide functionality of the system, are contact, client, Stay and employee. Following Figure 3.4 shows the classes and relationship with its multiplicity between them.

3.3. CLASS DIAGRAM

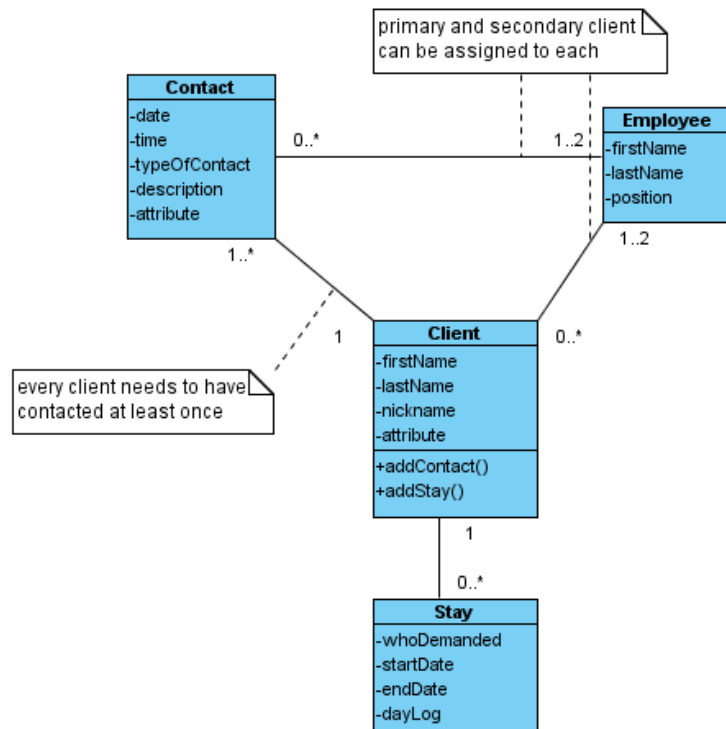


Figure 3.4: Class diagram – Analysis

Each client is associated with at least one contact. It would not make sense to keep a record of a client that has never contacted the centre. Each contact is associated with only one client. This is because even if more children come together, each report needs to show data about the single children, therefore the decision has been made to associate contact only with one client. Each client and contact can have up to two employees associated – primary and a secondary one.

These basic classes are the entity Java classes persisted in the database. Note that the number of attributes for each class is very large, therefore they are not entirely listed. Each attribute needs to have a getter and setter method, they are not noted in the diagram to keep it clear for the purpose of better understanding of the system.

3.3.2 Design

Design class diagram goes deeper than the analytical diagram. It also includes the classes of the entire system that provide certain functionality. This diagram shows `Home` that manages entities and `List` that provides the search functionality for selected criteria. They extend their respective Seam classes.

The implementation details using Seam framework is discussed in [Section 5.5](#).

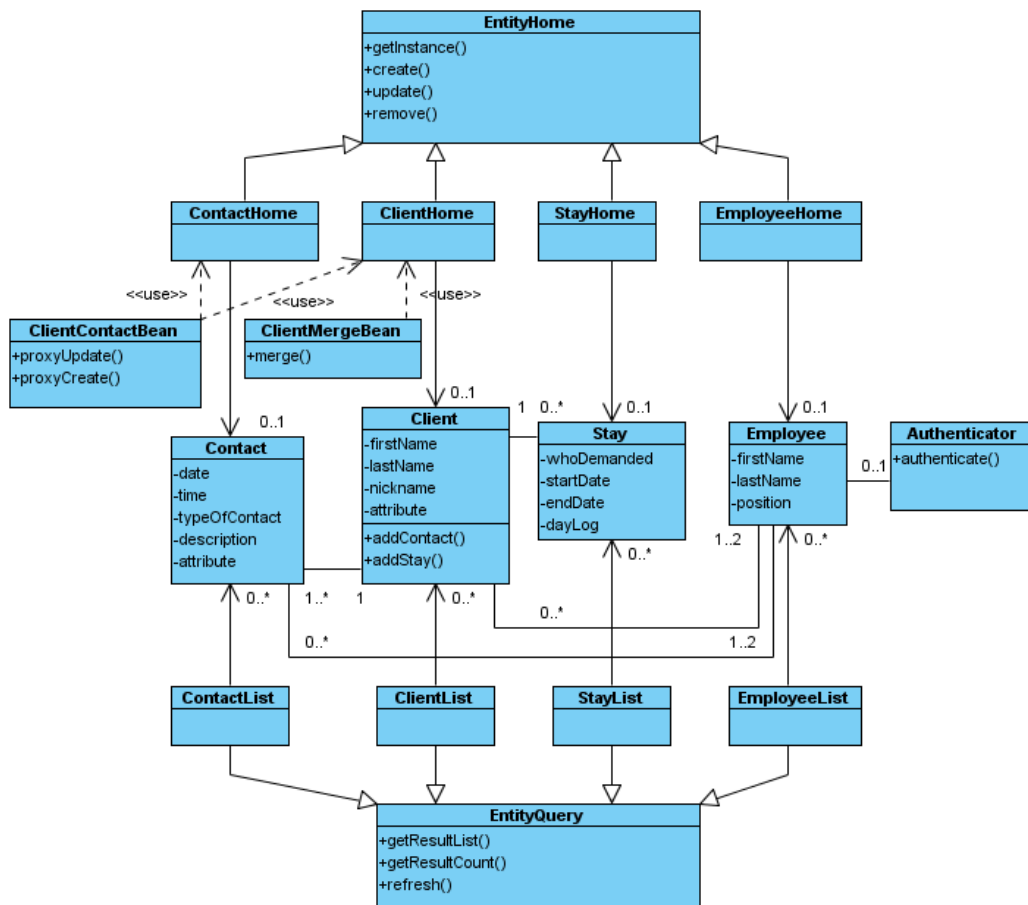


Figure 3.5: Class diagram – Design

3.4 Deployment diagram

Deployment diagram shows the software that is installed the physical hardware the system will be running on and possibly the middleware used to bridge the connection between separate parts. It is practical to create a deployment diagram for applications that are running on a network where more nodes are communicating together. It can also easily show what kind of data over which protocols is being sent in the network. [9]

Figure 3.6 shows a proposed structure of the deployment for the intranet application. The extent at which the diagram is modelled can be adjusted. It does not show the Java runtime environment, operating system or detailed persistence provider. It strives to show the selected technology. Employee's web browser connects over HTTP to the Apache web server which works as a load balancer which connects with JBoss nodes using AJP protocol

3.4. DEPLOYMENT DIAGRAM

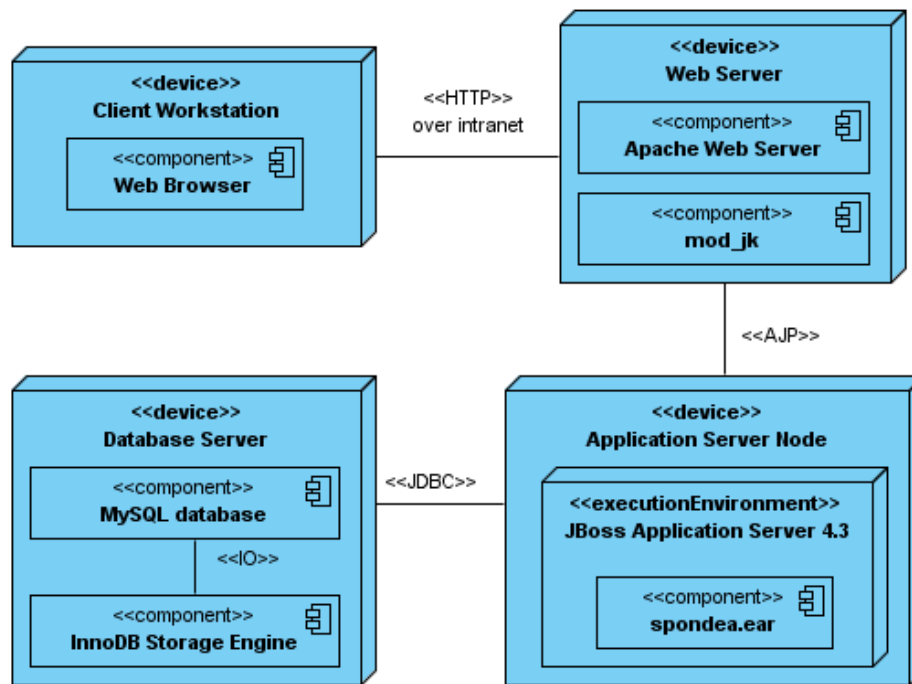


Figure 3.6: Deployment diagram (non-clustered)

supported by `mod_jk`¹ module. The database should be running on different node than application server and HTTP server.

To demonstrate clustering possibilities of selected technologies in [Section 4.6](#), the deployment diagram in a clustered environment has been done [Appendix B](#). Clustering a web application provides multiple benefits, most notably a performance gain, though it can eliminate a single point of failure or when a fail occurs, the client can failover² to a different node and keep working with his current session.

1. Documentation available at <http://tomcat.apache.org/connectors-doc/>

2. Failover is an ability to switch over automatically to a redundant server node upon a failure of the previously used node.

Chapter 4

Technology selection

4.1 Criteria

In this chapter I will explain the technology selection that has been done for the development, deployment and managing the information system. Object-oriented analysis in the previous chapter already puts certain constraints on the software. To select the technology objectively, it is important to summarize the criteria used in the selection process, which are described in the following section, and then defend the selected software that it fits the criteria.

4.1.1 Free and open sourced

Cost of software is one of the most important criteria when not dealing with mission critical and specialized solution for the problem domain. An information system and the needed supporting software for a non-profit organisation should not become a big expense. And this can be achieved nowadays keeping the same software quality as for paid software. This is possible due to open source approach to development which results in a fact that a modern enterprise-level technology can be used anywhere at virtually no cost.

Open Source software is generally distributed under a license that guarantees the right to read, modify, redistribute and use freely at no cost. Small open source projects are typically developed by a community of programmers interested in developing such software application for their specific purpose. However software companies may also develop an open source software solutions. To generate profit and make the development sustainable a company has to implement a business model for open source. One of the possibilities is by the subscription-based software. These companies will distribute their software for free and make their money from support contracts or develop customizations for the customer.

Purchase of a license of typical commercial software makes the solution dependent only on the software developer to add features or customize the software for the needs of your business or organization. The software manufacturer provides only the executable program without access to the source code.

Open source software has many advantages in this regard compared to the previous scenario. Using open source software enables you to freely modify the software and customize it in order to suit your application. This might be perceived as very demanding, time consuming and costly in the end. However, the open source community is usually made of

experts that develop plugins or bug fixes and publish them publicly for anyone to use instantly. One of the basic principles of open source is “release often”. This allows the software to be delivered to the user with all the latest issues resolved and with the latest technology.

Today’s situation in the open source software market is that such software is trusted and respected. This has been a big change in the IT business for past years. The developers of many of the open source solutions offer support contracts that are suitable to all levels of business or organization. Some of most successful and robust software on the web has been developed under such license.

Keeping open source and free software requirement in mind, the following criteria deal with open source as well.

4.1.2 Platform independent

Another important aspect of the software development is whether the solution is platform dependent or not. Even though the customer currently owns homogeneous IT infrastructure, it does not necessarily mean it sustain. At the time being, switch from proprietary Microsoft solutions to open, such as Linux, solutions is an actual trend since it lowers the total cost of ownership. To lower the cost (not only the time cost but also monetary) of possible transition, platform independence is suitable.

Whereas the platform independence typically means that the same executable in binary format (e.g. bytecode in Java terms) can be run on every platform that has a compatible runtime environment (e.g. JRE ¹ in Java terms), the term portability means that the code can be slightly rewritten and recompiled for the required platform and then run natively. Platform independence furthermore brings a major advantage that during development there is little or no regard needed on which specific platform the final solution will be deployed on. The same code can run on Windows, Linux, Solaris, etc. in their respective JREs.

To benefit of platform independence is noticeable, therefore it should be a criterium for software parts of the information system and especially of those which do not use open and standardized formats.

4.1.3 Robust, scalable and reliable

The load on the system in the first years or months of usage can be easily predicted. However later the load on the system might be out of the range that was expected. In that case, it is always important to choose robust and scalable solution right from the start. The initial deployment environment will certainly have a non-cluster setup, but it is important to keep in mind that a cluster setup might be deployed in future. It is also important to realise that there might be bottlenecks on multiple areas. To be able to cluster the entire solution, all parts that represent a possible bottleneck have to be scalable, e.g. the web server, the web container, the database.

1. Java Runtime Environment

4.1.4 Supported

Support does not have to necessarily mean that a company we have the software from provides over the phone or e-mail paid support. There are other ways of support to simply get problems resolved. For what will be needed a support from the community around the project is sufficient. If the community is large enough, it is possible to obtain faster responses than from the paid technical support, which is bound by the rules, unlike the community members that do not have affiliation with the company.

Community is a groups of developers and users that communicate through on-line discussion groups and other channels. Many open source projects run have an official community. Both the community and the software producer or vendor benefit from this system. This allows common problems to be easily solved and bugs to be quickly exposed and fixed and then later sell higher quality software to the enterprise.

4.1.5 Secure

The information system will be operating with confidential data. The confidentiality is ordered by the law and any kind of security breach can have negative influence on the operation of the centre. Even though it will be running solely in the intranet and therefore will be firewalled from the outside world, security threads from the internal network could employ a vulnerability in the software.

There is a general “myth” that the open source software is more vulnerable to an attack than proprietary solutions. Actually, the opposite is often true as experience shows over the years. Because the source code is exposed, it is often easier for a community of people around the project and especially those who are security minded to analyze and resolve potential security problems. Since open source has a faster release cycle, known issues are easier to resolve than in proprietary solutions.

4.2 Java Enterprise Edition 5

Java Enterprise Edition is a platform for developing modern, robust, scalable, transactional, secure and platform-independent applications written in object-oriented programming language Java. The key advantage of this platform is to provide a ready infrastructure and help minimize effort and resources needed to develop an enterprise application. Building on the standard version, Java Standard Edition known as Java SE, Java EE offers enterprise level features by provided APIs to simplify communication, database access, web development, etc.

The Java EE 5 platform is the newest version of Java EE at the time being. Improvements form the previous version include acceleration and radical simplification of Enterprise Java development, especially for Web Services (JAX-WS 2.0), Web Applications (JSF), and transactional components (EJB 3.0) and introduction of new database persistence model (Java Persistence) and using Java annotations.

The new Enterprise Java Bean technology, EJB 3.0 greatly simplifies the programming model through plain old Java objects (known as POJOs), which can easily be converted to web service just with annotations or made persistent to the database using the Java Persistence API. It is much easier to write a stateless and stateful components that take full advantage of the transactional capabilities of the EJB container with advanced transaction handling. [2]

The new Java Persistence is a much cleaner approach to mapping Java objects to relational databases. It benefits greatly from the ORM tools such as Hibernate, TopLink as well as Java Data Objects (JDO).

The improved JavaServer Faces 1.2 technology simplifies the building of user interfaces for web-based applications by providing pre-packaged components, significantly reducing new code development. The latest version offers better alignment with JavaServer Pages, improved state-saving behavior, and the ability to turn off component ID generation.

Java EE draws upon annotations and generics from Java SE basically eliminating the need for deployment descriptors. Annotations now allow declarative information to be moved inside the code and make it much easier to employ persistence, web services, transactions, security, and all the other powerful capabilities of Java EE. Having compatibility in mind, the previous approach without annotation is still possible and supported. [1]

All in all, the major enhancements in Java EE 5 noticeably reduce the amount of code and configuration with which application developers have to contend improving their productivity. Java EE 5 removes work of implementing reusable parts in creating applications and reduces the amount of time spent worrying about Java EE “plumbing” and therefore allows developers to concentrate more on the business logic while saving time.

4.3 Seam framework

Seam framework is an open source project of JBoss Inc., a division of Red Hat. Despite its name, the extend of it goes further than just a web framework. A better classification is that Seam is an application stack for Enterprise Java. It is inspired by several principles.

Seam defines a uniform component model for all business logic in the application. Seam components can be stateful, remembering the state already associated with defined contexts, including the long-running, persistent, business process and the conversation context, which is preserved throughout multiple web requests in a user interaction. There is no difference between components of presentation layer and business logic components in Seam. Applications can be structured according to one’s architectural needs, rather than having to stick with the unnatural logic scheme given by the combination of current frameworks. Compared to plain Java EE components, Seam components may simultaneously access state associated with the web request and state kept in transactional resources without having to propagate the web request programatically via method parameters.

JSF and EJB 3.0 are surely the two best new features in Java EE 5 as mentioned earlier but there is a minor flaw in the specifications. While JSF is a great component model for the presentation tier, EJB 3.0 is a new powerful component model for server side business and

persistence logic. JSF and EJB 3.0 function best when used together, however the Java EE 5 specification does not provide a standard way to integrate these two component models. By providing extension points to allow integration with other frameworks, Seam is able and does unify the component models, eliminating the need of code in between, and letting the developer only think about the business problem.

Seam supports the best open source JSF-based AJAX solutions RichFaces and ICEfaces. These ready solutions enable enhancement of user interface by AJAX capability without the need to write and even understand JavaScript code. Furthermore, Seam provides a JavaScript remoting layer that lets you call components asynchronously from client-side JavaScript without the need for an intermediate action layer. These rely on Seam's built-in concurrency and state management, which ensures that concurrent, asynchronous AJAX requests are handled correctly on the server side.

Optionally, Seam provides transparent business process management via jBPM which makes easy to implement complicated workflows, collaboration and task management. Seam also allows definition of presentation tier pageflow using JPDL, similar to BPEL, that jBPM uses for business process definition. With JSF's rich event model for the presentation tier, Seam can enhance this model further by exposing jBPM's business process related events via the same event handling mechanism, altogether providing a uniform event model for uniform component model.

There are many more features in Seam that build on the fundamentals. The most notable ones are:

- Bijection enhances dependency injection making it dynamic, contextual, and bidirectional. It acts as a mechanism for aliasing contextual variables to attributes of the component. It allows a component to safely and easily operate with value of a context variable by simply assigning it to an attribute of the component.
- Application using Seam allow the user to freely switch between multiple browser tabs each associated with a different conversation. They may take advantage of workspace management, enabling the user to switch between conversations in a single browser tab.
- EJB 3.0 takes advantage of annotations and "configuration by exception" as the fastest way to provide information to the container in a declarative form. Eliminating JSF's XML configuration files, Seam extends the annotations provided by EJB 3.0 by declarative state management, reduces the required XML to just that information which should be in XML, such as JSF navigation rules.
- To enable testing of web applications, Seam provides testability as core feature of the framework. Supported JUnit or TestNG tests that reproduce a whole interaction with a user can be written, exercising components of the system apart from the view. Tests can be run directly inside the development environment, where Seam will automatically deploy EJB components in embedded application server.

The need for a application stack like Seam should be clearer after understanding the arguments above. Today, there is more on a web application than serving simple HTML pages. To provide an easy interface to all this, Seam integrates JPA and Hibernate3 for persistence, the EJB Timer Service and Quartz for lightweight asynchronicity, JMS for messaging and JBoss Cache for page fragment caching, Meldware Mail for email, Hibernate Search and Lucene for full text search, jBPM for workflow, JBoss Rules for business rules. Seam structures an innovative rule-based security framework on JAAS and JBoss Rules. There are JSF tag libraries for rendering PDF, outgoing email, charts and wiki text. Seam components may be called synchronously as a Web Service, asynchronously from client-side JavaScript, Google Web Toolkit or directly from JSF. [7] [6] [5]

Seam can be freely downloaded from seamframework.org website ², or as a commercial and supported product it is bundled with JBoss Enterprise Application Platform (EAP), more information can be obtained at jboss.com website ³.

4.4 JavaServer Faces, Facelets

JSF or JavaServer Faces is technology developed through the standard Java community process under JSR - 314. With its arrival it establishes the standard for building server-side user web interface. The contributions from the expert group, the JavaServer Faces APIs are designed so that they can be leveraged by tools that will make web application development even easier, such as RichFaces. The primary goal being the design of easy to use presentation layer, the JavaServer Faces architecture clearly defines a separation between presentation and application logic while still keeping it easy to integrate those two.

Using JSF in JavaServer Pages (JSP) is quite common, however Seam developers recommend using Facelets as the view handler for JSF instead of JSP. Mainly because it provides an extensible templating system similar to Struts Tiles. Templating is a powerful concept together with JSF. Facelets is a lightweight view technology in presentation tier that is designed specifically for creating JSF pages. It accommodates JSF component markup natively and builds the UI component tree from it directly, avoiding the redundant JSP tag layer. Facelets templates use the same XML tags as JSP, but solves the core problem with JSP since it mismatches with the JSF life cycle and resulting noteworthy overhead. The ease-of-use for the developer, comes from allowed use of inline expression language (EL) notation without having to embed it in an `<h:outputText>` tag. These are just two small improvements that Facelets offers over JSP. In addition, Facelets offers a feature known as a composition, which is a fancy way of saying a reusable page fragment. To use Facelets, all that has to be done is to register the Facelets view handler in the `faces-config.xml` descriptor in the `WEB-INF` directory of the web application. [3] [4] [6]

More information about the JSF technology is available at JavaServer Pages Technology page at sun.com ⁴.

2. <http://www.seamframework.org/>

3. <http://www.jboss.com/products/platforms/application/>

4. <http://java.sun.com/javasee/javaxserverfaces/>

4.5 RichFaces

Seam framework comes with support for RichFaces. It is a rich component library for JSF and a framework for easily integrating AJAX capabilities into an interactive web application. The ready-to-use components come bundled in, so web developers can immediately take advantage of component features to create web applications that provide greatly improved user experience more reliably and more quickly. RichFaces supports skins for JSF applications. RichFaces also takes full advantage of the benefits of the JSF framework including validation, lifecycle and conversion support, with the management of static and dynamic resources.

RichFaces differ from other AJAX design architecture by enabling page-oriented AJAX support instead of the classical component-oriented support. It allows definition of an event in the page that invokes a AJAX request and then the option to define parts of the page that have to be updated according to the JSF component tree after the request is processed on the server.

The latest version of RichFaces at the time being is 3.3.0 which was released on January 13, 2009. More information about RichFaces technology can be obtained at RichFaces page at jboss.org⁵.

4.6 JBoss Application Server 5

In Java platform, the term application server refers to a Java Enterprise Edition application server. A generic application server, in a multiple-tier architecture, is a server that provides an API to access business logic and business processes to be used by third-party applications running on the server. Such design brings many advantages including data and code integrity by centralizing business logic on an single or few server machines, guaranteed atomic updates, transaction support as the server takes away the challenge from the developers who can focus on business logic relying on atomic transactions, centralized configuration, improved security by taking responsibility for authentication away from the potentially insecure client layer without exposing the database layer and last but not least performance gain by limiting the network traffic to other detached tiers.

JBoss Application Server is the most widely used Java application server today mostly developed by professional open source developers contributing, then productized and sold with support enabling it to keep its high standards. Since version 5 it is a Java EE 5 certified platform for development and deployment of enterprise Java and web applications providing full range of new Java EE 5 features along extended services including support for clustering and distributed caching and others.

JBoss is the best application server pick for project of this extend since it is well integrated, tested and supported with core solutions that were chosen previously. JBoss EAP is bundled with Seam framework. JBoss Application server can be downloaded at the com-

5. <http://www.jboss.org/jbossrichfaces/>

munity page jboss.org ⁶.

4.7 MySQL database

MySQL is the world's most popular open source database. Many of the largest and fastest-growing organizations chose MySQL to operate their high-traffic web sites, critical business systems and packaged software due its open source advantages but already building on its notable performance, reliability and benefits of ease of use.

Choosing MySQL for the project brings other advantages according to the specified criteria. Even though database server is usually a separate physical system, it can still be used on any supported major platform due to its portable versions and avoid vendor lock-in. Interoperability is guaranteed by its provided `Connector/J` which is the official JDBC driver for MySQL.

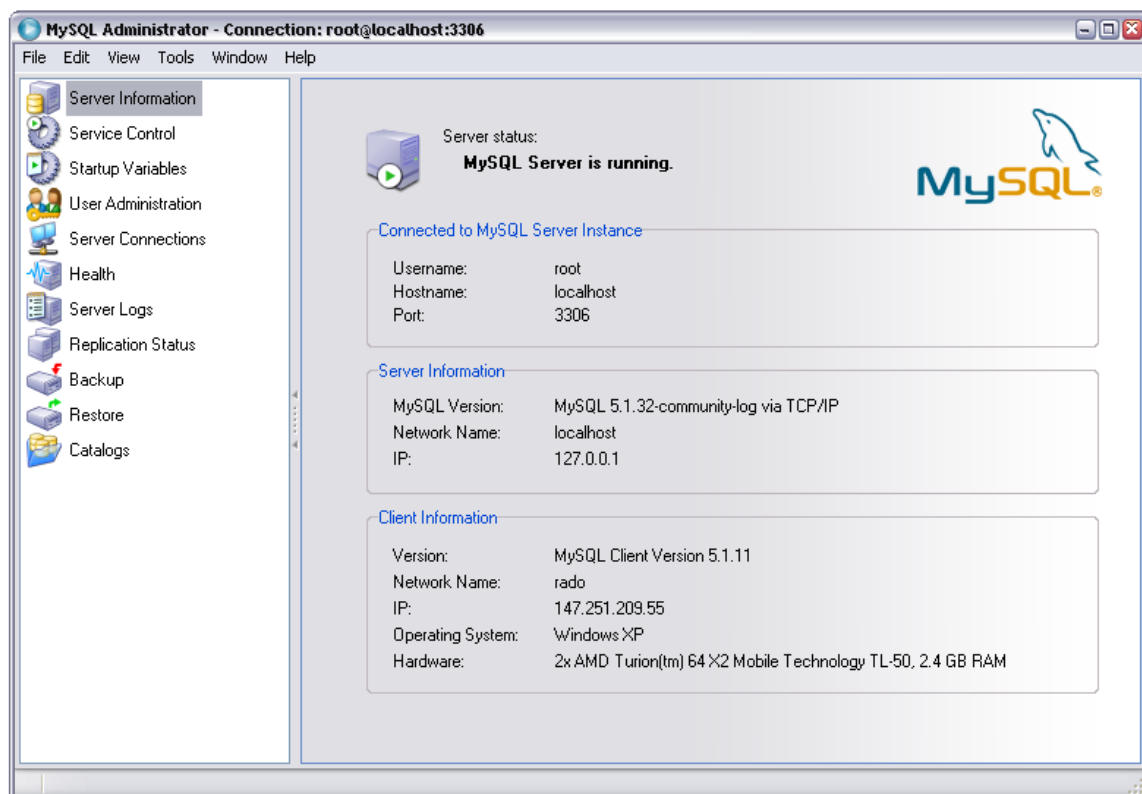


Figure 4.1: MySQL Administrator overview of the system and its feature set on the left

Furthermore, MySQL also provides a set of tools to manage the database. Working with confidential and important data requires certain level of data redundancy and data back-up.

6. <http://www.jboss.org/jbossas/downloads/>

MySQL Administrator as shown in Figure 4.1 allows the IT or information system administrator to easily and without further understanding of database architecture to plan a back up and restore data. MySQL Query Browser is another application, that allows the administrator to create and modify tables as shown in [Section 5.1](#) or simply browse and monitor the data with aided query creation.

MySQL was originally developed by MySQL AB with its first release in 1995, later in 2008 acquired by Sun Microsystems which is trying to bring its mature and evolved open source culture to the product and make it more enterprise ready but still enable community to use the software for free. The latest version is 5.1 and can be obtained from MySQL's official website ⁷.

4.8 Hibernate

Object-relational mapping (ORM) is a programming method for accessing data stored in relational databases comfortably in object-oriented programming language. Since the data in the relational database has different properties, like stateless unlike objects, than the objects in object-oriented language, a conversion is needed. Instead of directly querying the database, new instances of “virtual” objects are created and used within the programming language. Based on transactions, changes in the model can be committed into the database. Since the need for ORM tools is large, there are multiple free and commercial solutions available. Using a well-tested solution has a significant advantage compared to creating own custom tools, since its optimized.

Hibernate is a one of the ORM solutions. It is powerful, high performance persistence and query framework. Hibernate supports queries in its custom portable SQL extension known as HQL, in native SQL and an object Criteria and Example API. This allows applications to be portable since even though SQL is standardized, it cannot take advantages of specific SQL dialects resulting in incompatibility or performance issues. Hibernate strives to guarantee that the SQL in relational technology and knowledge is valid and optimized, by showing the result native SQL query to the user. Hibernate implements JPA used in EJB 3.0 containers.

The open source license allows the use of Hibernate and its companion projects in open source as well as commercial projects free of charge. Hibernate is a professional open source project and its developed to acts as a critical component in JBoss Enterprise Middleware System solutions. The latest stable version of Hibernate Core is 3.3.1.GA released on September 11, 2008 and can be obtained at hibernate.org website ⁸.

7. <http://www.mysql.com/>

8. <http://www.hibernate.org/>

4.9 Development environment

One of the unofficial requirements of the organization was to produce solution that would be open and allow easy update of the information system. Therefore emphasize was put on selecting development environment that can be used free of charge and easy to work with. Seam generates a project for Eclipse but it contains Ant script that can be opened in any IDE⁹ that supports free-form project based on such script. NetBeans is an advanced and very user-friendly IDE that can easily handle such project.

Use of version control system or code management system is important too keep revisions and history of changes or it can server as a back-up. Very often, reverting to previous version can be helpful, the need for versioning is crucial in development. The selection was again the open and well-tested solution Subversion (SVN). Subversion¹⁰ is an open source version control system with many command-line and graphical tools. It supports all basic version control needs as well as branching, tagging, merging, file locking, etc.

9. integrated development environment

10. <<http://subversion.tigris.org/>>

Chapter 5

Implementation of operational information system

In this chapter the development with Seam framework using `seam-gen` will be described and highlighted its main benefits for the developer.

5.1 Preparing SQL

In **Chapter 3, “Analysis and design of the information system”** the object-oriented model has been created. Some tools allow the diagrams to be directly converted to the ANSI-SQL script (query), however Visual Paradigm’s academic licence does not permit this which allows to demonstrate the MySQL’s graphic user interface (GUI) tools.

MySQL Query Browser is an application released under open source licence to the community, ported to all major platforms – Windows, Linux and Mac OS. It can be freely downloaded from MySQL website’s download section ¹. After logging to a running database, the GUI allows running queries, executing scripts, editing data through the monitor, but mainly at the beginning of the development, creating and designing databases. (another tool from MySQL to create and design databases is WorkBench).

MySQL supports multiple storage engines developed by different engine developers. For simple website and storage MyISAM is the most popular choice. This is because it has excellent performance for simple select operations. This is also supported by the fact that this engine does not support transactions, constraints and foreign key references. However, for making the database manageable it is needed to have transactional model when dealing with multiple tables referencing each other. The recommended and shipped with MySQL distribution database engine supporting required features is InnoDB currently developed by Innobase ².

InnoDB provides full support of standard SQL isolation levels for ACID-compliant transaction processing, and maximizes throughput with row level locking and multi-version concurrency control (MVCC). InnoDB ensures high performance, minimizing expensive disk I/O, via many techniques to efficiently use memory and processor resources. InnoDB ensures data integrity and reliability with such capabilities as referential integrity support and automatic data recovery following hardware and software failures. InnoDB fully integrates with MySQL and supports such features as replication, stored procedures, triggers

1. <<http://dev.mysql.com/downloads/>>

2. <<http://www.innodb.com/>>

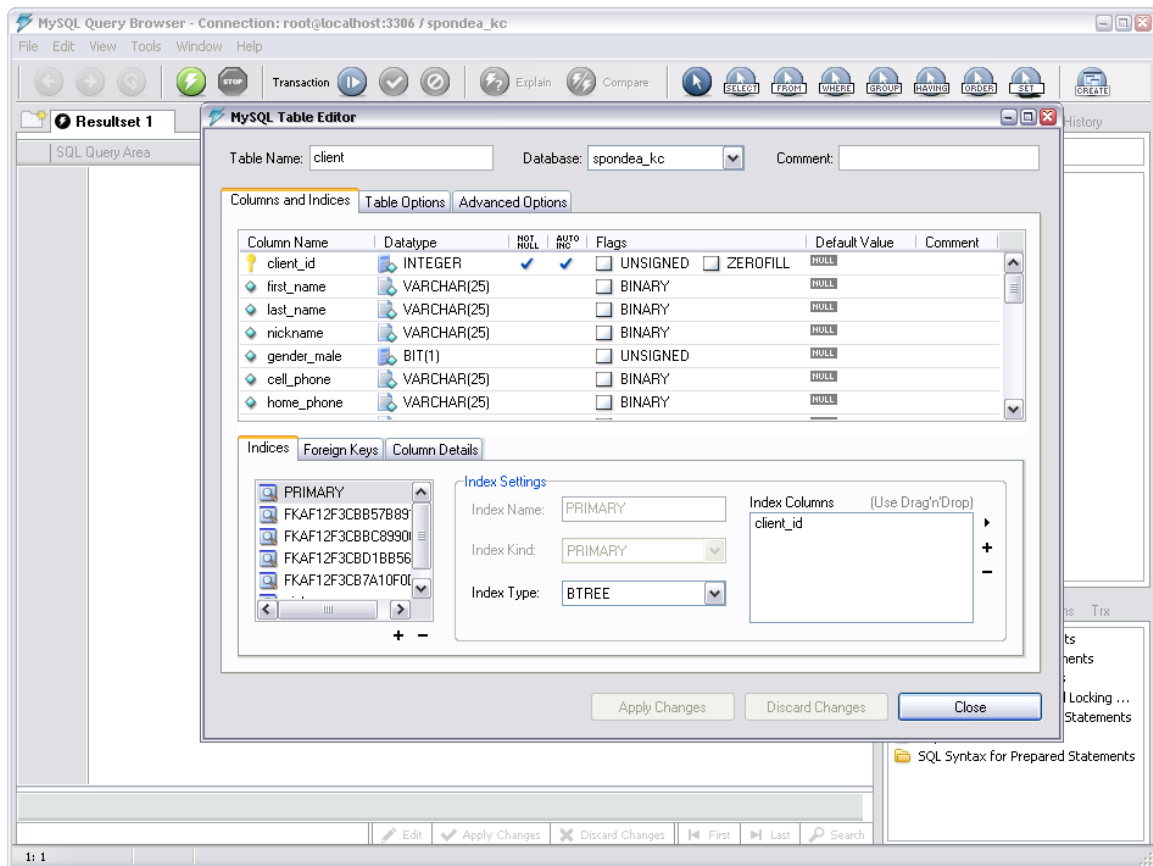


Figure 5.1: MySQL Query Browser editing table for entity Client

and views.

Looking at the types supported by the storage engine, they do not map one-to-one to the used Java types. While this could represent a time-consuming problem in other paradigms, JDBC and ORM make the differences transparent to the developer which also allows an easy switch from one database to other supported database by the ORM tool.

Designing databases, especially data types for columns should keep in mind that the rows will be handled with in objects. To enable best mapping back and forth and a best performance on both sides, its beneficial to choose data types that have a similar or exactly same representation. Enumerations were avoided, since its default mapping by Hibernate is a string and therefore still would have to be localized. For the database, the following mapping InnoDB to Java data types was used:

- `VARCHAR(<limit>)` or `TEXT` maps to `String`
- `TINYINT(3) UNSIGNED` represents `Enum`

- `BIT(1)` maps to `Boolean`
- `BIT(1) NOT NULL` maps to `boolean`
- `DATE` maps to `Date` without time annotated `@Temporal(TemporalType.DATE)`
- `DATETIME` maps to `Date` annotated `@Temporal(TemporalType.TIMESTAMP)`

SQL also provides stored procedures running within the database container and called from the application layer. Build-in functions are not going to be used because of several reasons which I will explain. Stored procedures could provide several benefits. They supposedly run faster than the single set of queries, however that is not always valid on non-enterprise databases and they actually run slower than the queries alone. Stored procedures can be used to ensure data integrity and consistency, however when transactions are supported and used by the ORM tools and other containers used, data integrity is guaranteed. Furthermore, it shifts or rather splits application logic to the database and the application logic layer. Such situation does not provide any benefits but also makes the entire solution less platform independent since the stored procedures are not database independent and have to be ported.

The created SQL scripts are part of the Bachelor thesis, see [Appendix A](#) for details.

5.2 Using seam-gen

The Seam framework package (at the time being, available production-ready version is 2.1.1.GA and a preproduction version 2.1.2.CR2 is available including new features in `seam-gen` such as generating identity management) includes tooling to set up a project from the beginning - create Eclipse project files and build script which can be used in conjunction with any IDE that has support for Ant scripts, reverse engineer existing database tables into entities and create sample create, read, update and delete (CRUD) application.

Executing `./seam setup` from the Seam directory initiates a simple Ant script that gathers data about the project to be created as shown in the following figure. By default, JBoss 4.2.3.GA is used but JBoss 5.x.GA is also possible.

The project can be deployed as a web application archive (WAR) or as an enterprise archive (EAR). Selecting WAR suppresses the possibility to use EJB components. Keeping in mind that Seam components can be easily annotated and used as EJB components, possible further feature development including a remote client is simplified compared to simple web archive.

By default three Java packages are created. For entities from the database, session beans and test cases. Entities do not support hot-deployment as well as session beans when working with EAR.

Following figure shows database setup. Supplying a JDBC driver, any supported database can be used. At the end, the property file is created which is used afterwards to create a project.

5.3. BOTTOM-UP AND TOP-DOWN

```
$ ./seam setup
Using seam-gen sources from: ~/jboss-seam-2.1.1.GA/seam-gen
Buildfile: ~/jboss-seam-2.1.1.GA/seam-gen/build.xml

init:

setup:
    [echo] Welcome to seam-gen :-)
    [input] Enter your Java project workspace (the directory that contains your
Seam projects)
~/Projects
    [input] Enter your JBoss AS home directory
~/jboss-4.2.3.GA
    [input] Enter the project name
spondea
    [echo] Accepted project name as: spondea
```

Figure 5.2: Snippet from command-line utility

```
[input] Is this project deployed as an EAR (with EJB components) or a WAR
(with no EJB support) ([ear], war)
ear
    [input] Enter the Java package name for your session beans
com.radoslavhusar.spondea
    [input] Enter the Java package name for your entity beans
com.radoslavhusar.spondea.entity
    [input] Enter the Java package name for your test cases
com.radoslavhusar.spondea.test
```

Figure 5.3: Snipped showing options of different deployment types

The project is created and files are copied executing `./seam create-project` command. This generates a simple Seam project skeleton that can be opened in IDE. To continue code generation, `./seam generate-model` uses Hibernate tools to analyze the database and map it to the entities created in the specified package. To generate new CRUD pages and controllers execute `./seam generate-ui` which creates xhtml pages and Java beans. An alias can be used to generate model and then user interface `./seam generate` which executes both generation targets.

5.3 Bottom-up and top-down

Bottom-up and top-down are two different development scenarios differing in which layer is developed first – the database or the entity classes. If the database schema already exists the development is called bottom-up. As was shown in the previous section, Seam's tooling

5.3. BOTTOM-UP AND TOP-DOWN

```
[input] What kind of database are you using? (hsql, [mysql], oracle,
postgres, mssql, db2, sybase, enterprisedb, h2)
mysql
[input] Enter the Hibernate dialect for your database
org.hibernate.dialect.MySQLDialect
[input] Enter the filesystem path to the JDBC driver jar
~/jboss-4.2.3.GA/server/default/lib/mysql-connector-java-5.1.7-bin.jar
[input] Enter JDBC driver class for your database
com.mysql.jdbc.Driver
[input] Enter the JDBC URL for your database
jdbc:mysql:///spondeo_kc
[input] Enter database username
spondeo_kc
[input] Enter database password
password
[input] Are you working with tables that already exist in the database?
y
[input] Do you want to drop and recreate the database tables and data in
import.sql each time you deploy? (y, [n])
n
[delete] Deleting: ~/jboss-seam-2.1.1.GA/seam-gen/build.properties
[propertyfile] Creating new property file:
~/jboss-seam-2.1.1.GA/seam-gen/build.properties
[echo] Installing JDBC driver jar to JBoss AS
[echo] Type 'seam create-project' to create the new project
```

Figure 5.4: Snippets showing database setup

provides a reliable way to generate Java entity classes from the existing database schema. While implementation is in progress, the attention is solely on the Java code and database becomes more transparent to the developer. It is important to have a way to apply changes in the entity classes to the database. This is part of the top-down development, when changes in the classes are reflected in the database layer. This is handled by Hibernate. To tell Hibernate to update the tables, it is necessary to set a property in `META-INF/persistence.xml` to update. [5]

```
<property name="hibernate.hbm2ddl.auto" value="update"/>
```

Obviously this does not make sense when the application is deployed in production and the property needs to be set up to validate that the table structure is the same as expected by the Java entity classes. The value could be set up as follows.

When major changes are done to the database layer, `seam-gen` can be used again to generate entities and replace the deprecated ones.

```
<property name="hibernate.hbm2ddl.auto" value="validate"/>
```

5.4 Development with Seam

Developing a web application with Seam exploits JBoss application server's functionality to perform hot deployment. It is a form of dynamic reloading when only few parts of the web application are changed and need to be viewed or tested by the developer. When developing an EAR type of project, the exploded EAR (which is a directory named same as an EAR file) can hot deploy only view pages. The command

```
./seam explode
```

searches for modified, added or deleted files and updates only those on the application server. To notify the server that the Java classes have changed as well, it is necessary to execute

```
./seam clean restart
```

which removes the previous application data and cache, undeploys the applications and deploys it again. The time difference is 1 second compared to about 8 seconds, respectively for explode and restart methods. This enables developer to quickly get a view on the application and perform manual tests.

However, repeated deployment sometimes leads to unused data stored in the heap which is not cleared by the garbage collector due to software bugs in the Java Virtual Machine or the application server. It is recommended on workstations with sufficient memory to set it to allocate more memory for the heap.

```
-Xms512m -Xmx1024m -XX:PermSize=256m -XX:MaxPermSize=512m
```

Additional garbage collector preferences can be turned on when experiencing out-of-memory issues. For most workstations the above setup should be sufficient for notable number of hot deployments. [5]

5.5 Seam components

This section tries to provide an insight to developing with Seam. The variety of possibilities is very large, therefore the main principles will be described.

A components are stateful objects managed by Seam container. There are no restrictions put on the object, thus it does not need to extend or implement a specific class, it can be just a simple Plain Old Java Object (POJO). Seam container handles creation instances and wraps them with interceptors which provide bijection and other functionality. All seam components have to be annotated by `@Name` annotation (or by XML descriptors), specifying a name which will then be used throughout the project to reference this component. [7] [6]

5.5.1 Home object

The first type, a Home object, provides persistence operations for a particular entity class. This enables the developer to be able to execute basic persistence operations without having `EntityManager` created or obtained manually. For a simple entity class from Figure 5.5 it will be shown how to create a home for it.

```
@Entity
public class Client {

    @Id
    private Long id;

    private String firstName;
    private String lastName;

    // Etc...
}
```

Figure 5.5: Example entity class

The class needs to be defined to be injected into the context. It can be defined via XML configuration. However, this is not used in the project, since its Seam's advantage that annotations can be often used instead of multiple configuration files.

```
<framework:entity-home name="clientHome" entity-class="Client" />
```

Figure 5.6: EntityHome defined via XML configuration

In Figure 5.7 `ClientHome` is defined and annotated by `@Name("clientHome")` to register this component with Seam container.

```
@Name("clientHome")

public class ClientHome extends EntityHome<Client> {
    ...
}
```

Figure 5.7: EntityHome defined via annotation

This really simplifies using the bean on the web, which can be can used directly from a JSF page as shown in Figure 5.8. A home object immediately provides operations to persist, remove, update and get instance which can be called from

```
<h:commandButton action="..." />
```

```

<h:form>
  <div>
    First name: <h:inputText value="#{clientHome.firstName}"/>
  </div>
  <div>
    <h:commandButton value="Create client"
      action="#{clientHome.persist}"
      rendered="#{!clientHome.managed}"/>
    <h:commandButton value="Update client"
      action="#{clientHome.update}"
      rendered="#{clientHome.managed}"/>
  </div>
</h:form>

```

Figure 5.8: Executing actions from a Home within JSF page

as well.

The generation of the page and the templating options with JSF and Facelets using expression language (`{expression}`) are very powerful. For instance, having a Java code decide whether to show create of an update button on the page, by checking whether the entity is already managed or not, would be way more complicated. Simply a home component can be used and method that returns whether the entity is managed or not can be inserted into `rendered` attribute.

There are four basic home components in the project, `ClientHome`, `ContactHome`, `StayHome` and `EmployeeHome`. They have several methods added and overridden to provide convenience methods.

5.5.2 Query object

Query objects do not provide updating the data unlike Home objects. They offer a convenient way to query the database tables while feeding data into restrictions. The Query objects extend `EntityQuery` class. Figure 5.9 shows an example of a Query object and Figure 5.10 shows how to obtain a result list in JSF using expression language.

```

@Name("clientList")
@Scope(ScopeType.CONVERSATION)
public class ClientList extends EntityQuery<Client> {
  ...
}

```

Figure 5.9: Definition of a `clientList` class extending `EntityQuery`

Having a scope set to `CONVERSATION`, it is possible to keep the search criteria when user navigates away from the page but does not end the conversation. This strives to provide

more comfort to the user. The search restrictions can be easily reset by ending the conversation.

```
<rich:dataTable value="#{clientHome.resultList}" var="_client">
  <rich:column sortBy="#{_client.firstName}">
    <f:facet name="header">
      <h:outputText value="#{messages.firstName}" />
    </f:facet>
    <h:outputText value="#{_client.firstName}" />
  </rich:column>
</rich:dataTable>
```

Figure 5.10: RichFaces table displaying data from `clientList` component

Figure 5.10 shows how entity can be obtained from the list and create a table. The example uses RichFaces' ³ `dataTable` element. The `EntityQuery` provides methods to obtain the results as well as support for pagination and ordering. The example is to show, that when a user uses the sorter, it will not call the `EntityQuery` ordering method but will use RichFaces table element to handle the sorting. This could only be done with small results sets to maintain a sufficient performance but at the same time, do not poll the database.

There are four basic Query components in the project: `ClientList`, `ContactList`, `StayList` and `EmployeeList`. Their implementations were extended to provide additional functionality and provide sorting on the first call.

3. Namespace "rich" was defines as `xmlns:rich="http://richfaces.org/rich"`

Chapter 6

Conclusion

The main objective of this thesis was to analyze, design and develop a functional prototype of an information system for a crisis centre. Analysis of the current state of information technology including the requirements capturing for the new system was done at crisis and intervention centre Spondeo based in Brno. The specification of requirements resulted in object-oriented analysis of a proposed information system. Keeping in mind the technology to be used for development, appropriate approaches to system design and development were followed. It is interesting to experience in real projects how analysis becomes helpful during implementation when the focus is on the code and technology, however analysis still provides a “guide” on how to continue and what has to be done.

One of the aims of the thesis was also to explore the latest modern technology and make appropriate software selection. As a result, chosen Java EE 5 platform with Seam framework and the dependent technology have proven that they are suitable for development of rich web applications. Starting with rapid development using Seam’s tooling, Java Server Faces technology’s extensible component model enabling RichFaces to effortlessly bring visually pleasing components as well as providing the developer with set of high-level interfaces to manage their states. Using Facelets rapidly boosts user interface templating possibilities compared to Java Server Pages technology, making the presentation layer more manageable.

The most interesting outcome of this Bachelor thesis is the implementation of a prototype of the information system. The prototype is fully functional, although it is only a beginning of it as a project. After deployment to production, the users will only start to realize what other features an information system provide and can be further extended and the interface can be optimized for the most frequently executed operations. The architecture is very modular and easy to extend, so any suggestions could be implemented.

My personal hope is that this project could eventually become a free and open source software available for all kinds of similar organizations, free to download, extend and share. It is interesting to see how intangible software can improve productivity and in this case potentially help more children who are in need.

Bibliography

- [1] *Java Platform, Standard Edition 6: API Specification*, Sun Microsystems, Inc., <<http://java.sun.com/javase/6/docs/api/>> , 2006. 4.2
- [2] *Java EE FAQ*, Sun Microsystems, Inc., 2009, <http://java.sun.com/javaee/overview/faq/javaee_faq.jsp> . 4.2
- [3] *JavaServer Faces Technology*, Sun Microsystems, Inc., 1994-2009, <<http://java.sun.com/javaee/jaserverfaces/>> . 4.4
- [4] Dudney, B. and Lehr, J. and Willis, B. and Mattingly, L.: *Mastering JavaServer Faces*, Wiley Publishing, Inc., 2004, 0-471-46207-1. 4.4
- [5] Allen, D.: *Seam in Action*, Manning Publications, June 15, 2008, 978-1933988405. 4.3, 5.3, 5.4
- [6] *Seam 2.1.1.GA Reference*, <<http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/html/>> , December 15, 2008. 4.3, 4.4, 5.5
- [7] Juntao Yuan, M. and Heute, T. and Orshalick, J.: *Seam Framework: Experience the Evolution of Java EE*, Prentice Hall, February 14, 2009, 978-0137129393. 4.3, 5.5
- [8] Arlow, J.: *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky (UML 2 and the unified process)* , Computer Press, 2007, 978-80-251-1503-9. 3.2, 3.3
- [9] *Visual Paradigm Resource*, Visual Paradigm, March 29, 2007, <<http://resource.visual-paradigm.com/>> . 3.1, 3.4

Index

bijection, 16

candidate classes, 6

database, 9

deployment diagram, 10

EAR, 27

EJB, 15

Facelets, 17

Hibernate, 20

JavaServer Faces, 17

JBoss AS, 18, 27

MySQL, 19, 22

object-oriented analysis, 12

open source, 12

ORM, 20

persistence, 15, 26

portability, 13

RichFaces, 18

Seam, 9, 16, 18, 22, 24, 27

Spondeia, 1, 3, 6

Textual analysis, 6

UML, 6, 7

use case diagram, 7

Appendix A

Contents of the attached CD

install/

a bundle containing a final deployed information system bundled in JBoss Application Server 4.3

spondeia/

sources of the information system

sql/

SQL script used for bottom-up development and a final optimized script

guide/

directory containing a quick user and administration guide for the information system

test/

Selenium tests used for functional and performance testing

thesis/

directory containing the PDF version of this document as well as source files needed to compile it

uml/

Visual Paradigm for UML Standard Edition (download from <<http://www.visual-paradigm.com/>>) project file including all UML diagrams used in this Bachelor thesis. Created using academic licence for Masaryk University.

Appendix B

Deployment diagram

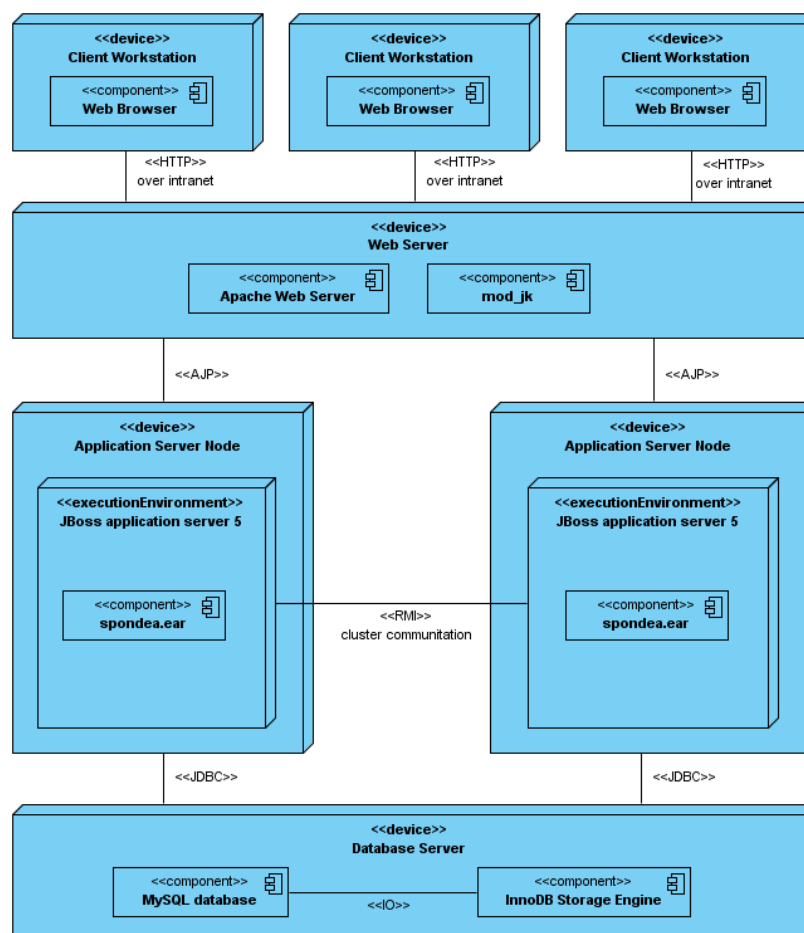


Figure B.1: Deployment diagram with cluster set-up