Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science and Engineering

Master thesis

PREPARATION OF ON-LINE ASSESSMENTS

Josef Bicik


Prague, January 2006
České Zadání
English Assignment Form
Anotace

Tato diplomová práce se zabývá ověřováním znalostí studentů v e-vzdělávání pomocí internetu. V první části je čtenář uveden do problematiky testování a je provedena analýza několika existujících produktů, které se používají v e-vzdělávání, se zaměřením zejména na webové testování. Výsledkem analýzy je specifikace ideálního testovacího systému pro tvorbu a distribuci online testů. Druhá část popisuje návrh, implementaci a výsledky testování jednoduchého webového testovacího systému s názvem I-Test.
Abstract

This master thesis deals with assessment of students in e-learning using the Internet. The first part is an introduction to the basics of the theory of assessment and contains an analysis of several e-learning systems aimed at web testing. The analysis results in specification of ideal assessment system for creation and delivery of online assessments. The second part describes design, implementation and results of the testing of a simple web-testing system named I-Test.
Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 20. 1. 2006 .................................................
podpis
Acknowledgments

I would like to thank all who contributed to the successful completion of this master thesis. I am most grateful to my supervisor Ing. Bozena Mannova, M.Math, for her valuable remarks, insight and patience. I also thank Jan Rejthar - the manager of implementation of e-learning at CTU for providing access to Microsoft Class Server and Dr. Michal Pavlo, headmaster of Skolni Primary School for allowing the testing of the application on his computer cluster. Finally, I thank my parents and all my friends for their support throughout the creation of this thesis and the course of my studies.
# Table of Contents

1 Introduction ................................................................................................................... 25  
1.1 The Importance of Assessment in e-learning ........................................................... 25  
1.2 Objectives of the Thesis ........................................................................................... 26  

2 The Theory of Assessment ............................................................................................ 27  
2.1 Types of Assessment Items ...................................................................................... 27  
2.1.1 Closed Items ..................................................................................................... 27  
2.1.2 Open Items ....................................................................................................... 28  
2.2 Issues Related to Multiple-choice Items .................................................................. 29  
2.2.1 Grammatical Cues ............................................................................................ 29  
2.2.2 Long Correct Answer ....................................................................................... 29  
2.2.3 Unintended Clue ............................................................................................... 30  
2.2.4 Ambiguity ......................................................................................................... 30  
2.3 Quality Factors of an Assessment ............................................................................ 30  
2.3.1 Reliability ......................................................................................................... 30  
2.3.2 Sensitivity ......................................................................................................... 31  
2.3.3 Difficulty .......................................................................................................... 31  
2.3.4 Objectivity ........................................................................................................ 32  
2.3.5 Validity ............................................................................................................. 32  

3 Analysis of Existing e-learning Systems ...................................................................... 33  
3.1 E-learning Standards ................................................................................................ 34  
3.1.1 IMS ................................................................................................................... 34  
3.1.2 SCORM ............................................................................................................ 36  
3.2 MS Class Server ....................................................................................................... 38  
3.2.1 User roles in Microsoft Class Server ................................................................ 38  
3.2.2 Assessment Possibilities Offered by Microsoft Class Server ......................... 39  
3.3 MOODLE ................................................................................................................. 39  
3.3.1 Assessment Possibilities Offered by MOODLE .............................................. 40  
3.4 ATutor ..................................................................................................................... 41  
3.4.1 Assessment Possibilities Offered by ATutor ................................................... 42  
3.5 Ilias ...................................................................................................................... 42  
3.5.1 Assessment Possibilities Offered by Ilias ........................................................ 43  
3.6 Questionmark Perception ......................................................................................... 43  
3.7 Hosted Test ............................................................................................................... 44  
3.8 Outcomes of the Analysis ........................................................................................ 45  
3.8.1 Key Features of an Assessment System ........................................................... 46  
3.8.2 Key Attributes of an Online Assessment ......................................................... 46  
3.8.3 Supported Assessment Items ............................................................................ 47  
3.8.4 Summary .......................................................................................................... 47  

4 Design ............................................................................................................................ 49
List of Figures

Fig. 3.1 Structure of IMS Package ................................................................. 35
Fig. 3.2 Participants of content exchange and their collaboration ...................... 36
Fig. 4.1 Zaval Java Resource Editor [18].......................................................... 52
Fig. 4.2 MVC architecture ............................................................................. 56
Fig. 4.3 Difference between the use of Framework and Library ....................... 57
Fig. 4.4 Layout of web pages generated by I-Test ........................................... 62
Fig. 5.1 Part of simplified page flow diagram ................................................... 63
Fig. 5.2 Context diagram of I-Test system ....................................................... 64
Fig. 5.3 Use case diagram of Administrator .................................................... 65
Fig. 5.4 Use case diagram of Test Manager ..................................................... 66
Fig. 5.5 Use case diagram of Student .............................................................. 66
Fig. 5.6 Class Diagram of I-Test in UML ......................................................... 66
Fig. 5.7 E-R diagram of I-Test ...................................................................... 66
Fig. A.1 Creation of new user account in pgAdmin ........................................... 66
Fig. A.2 Creation of new database in pgAdmin ................................................. 66
Fig. A.3 Setting properties of new database ..................................................... 66
Fig. B.1 Initial login screen of I-Test ............................................................... 66
Fig. B.2 User details and settings ................................................................. 66
Fig. B.3 List of Test Managers ................................................................. 66
Fig. B.4 Adding or editing Test Manager ..................................................... 66
Fig. B.5 Access log ................................................................. 66
Fig. B.6 Importing students ................................................................. 66
Fig. B.7 Adding or editing a question ............................................................ 66
Fig. B.8 Creating or editing an assessment ................................................. 66
Fig. B.9 Delivering an assessment ............................................................... 66
Fig. B.10 Test statistics ................................................................. 66
Fig. B.11 List of available assessments ......................................................... 66
Fig. B.12 Warning message displayed before submission of the test ............... 66
Fig. B.13 Sample assessment with timer ....................................................... 66
Fig. B.14 Completed assessment with correct answers ................................. 66
# List of Tables

Table 5.1 Properties of class User ............................................................................................ 66
Table 5.2 Properties of class Student ....................................................................................... 66
Table 5.3 Properties of class TestManager .............................................................................. 66
Table 5.4 Properties of class TestInstance ............................................................................... 66
Table 5.5 Properties of class Test ............................................................................................. 66
Table 5.6 Properties of class Question ..................................................................................... 66
Table 5.7 Properties of class QuestionBank ............................................................................. 66
Table 5.8 Properties of class TestInstanceQuestion ................................................................. 66
Table 5.9 Properties of class Answer ....................................................................................... 66
Table 5.10 Properties of class Access ...................................................................................... 66
Table 6.1 Configuration of laptop used for testing ................................................................. 66
Table 6.2 Sample assessment statistics .................................................................................... 66
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADL</td>
<td>Advanced Distributed Learning</td>
</tr>
<tr>
<td>AICC</td>
<td>Aviation Industry CBT Committee</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ARIADNE</td>
<td>is a foundation which exploits and further develops the results of the ARIADNE and ARIADNE II European Projects</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>CTU</td>
<td>Czech Technical University</td>
</tr>
<tr>
<td>DAO</td>
<td>Data Access Object</td>
</tr>
<tr>
<td>DHTML</td>
<td>Dynamic HTML</td>
</tr>
<tr>
<td>DOD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise Java Beans</td>
</tr>
<tr>
<td>GPL</td>
<td>General Public Licence</td>
</tr>
<tr>
<td>HQL</td>
<td>Hibernate Query Language</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IMC</td>
<td>Internet Mail Consortium</td>
</tr>
<tr>
<td>IMS</td>
<td>Instructional Management Systems</td>
</tr>
<tr>
<td>J2EE</td>
<td>Java 2 Platform, Enterprise Edition</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>JSP</td>
<td>Java Server Pages</td>
</tr>
<tr>
<td>K-12</td>
<td>North American designation for primary and secondary education</td>
</tr>
<tr>
<td>LCMS</td>
<td>Learning Content Management System</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management System</td>
</tr>
<tr>
<td>LO</td>
<td>Learning Object</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MVC</td>
<td>Model View Controller</td>
</tr>
<tr>
<td>ORM</td>
<td>Object-Relational Mapping</td>
</tr>
<tr>
<td>PHP</td>
<td>PHP: Hypertext Preprocessor</td>
</tr>
<tr>
<td>PIF</td>
<td>Package Interchange File</td>
</tr>
<tr>
<td>POJO</td>
<td>Plain Old Java Object</td>
</tr>
<tr>
<td>QTI</td>
<td>Question and Test Interoperability</td>
</tr>
<tr>
<td>SCO</td>
<td>Sharable Content Object</td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Content Object Reference Model</td>
</tr>
<tr>
<td>SIF</td>
<td>School Interoperability Framework</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>UNICODE</td>
<td>Multi-byte encoding developed by Unicode consortium capable of coding characters of all languages from the whole world for use in computers.</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WAR</td>
<td>Web Application Archive</td>
</tr>
<tr>
<td>WCAG</td>
<td>Web Content Accessibility Guidelines</td>
</tr>
<tr>
<td>XHTML</td>
<td>eXtensible HyperText Markup Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>ZIP</td>
<td>popular data compression and archival format</td>
</tr>
</tbody>
</table>
1 Introduction

E-learning is a modern and rapidly developing education process. Its importance has significantly grown with the expansion of the Internet and improved performance of personal computers nowadays. The greatest benefit of e-learning is being able to receive education regardless of proximity and distance between teacher-student. However, the demands of people are constantly changing. Hence, new learning methods must be developed such that it copes with the changes. To children and students with high expectations and requirements, e-learning offers an attractive and effective way of educating by using interactive learning materials which involves more use of the senses than typical classroom learning methods. Thus, e-learning helps students retain newly acquired knowledge more easily. The benefits of e-learning combined with those of traditional teaching methods form a modern type of education known as blended learning.

Not only are people evolving, e-learning has also changed dramatically since its very beginning. It has developed from simple applications used to assess the knowledge of students many years ago to complex e-learning systems offering many interesting features. E-learning nowadays is mostly based on robust Learning Management Systems. Such systems are often integrated within information systems of educational institutions and simplify the whole administration process linked to teaching and assessing of students.

Assessment modules are becoming a common part of all major e-learning systems. In case they are stand-alone applications, they offer functionality allowing their integration within those systems to further extend their capabilities. Testing of the knowledge of students using the Internet or intranet allows institutions to deliver tests to large amounts of students with minimum additional costs. This kind of testing facilitates storage of test results and their further analysis. It also eliminates time required for grading simple quiz questions which can be scored automatically and hence does away with the human factor normally involved which reduces the probability of making a mistake hence guaranteeing high objectivity.

1.1 The Importance of Assessment in e-learning

The need to assess the progress of students comes side by side with teaching. Assessment is an excellent feedback for teachers so that they know if their students have managed to understand the topic and gained the required level of knowledge. According to the results of assessment the teacher can focus on specific problematic parts of the curriculum or even
better, address problems of individual students. In the past, analyses of the students’ results were very tedious and difficult. Modern e-learning systems usually provide all important statistics with no additional costs and thus save a lot of time. The importance of these measures will be discussed in chapter 2.3.

Teachers are not the only ones who need assessments to monitor progress of their students. In learner-led education, e-learning systems help learners to assess their own progress and according to the results allow them to move on to another chapter. Although the main task of assessments in most cases is to assess acquired knowledge they can be also used to promote learning and reduce forgetting.

Testing plays a very important role in adaptive e-learning systems. Such systems compose the content of the course from Learning Objects (LOs) according to the progress of the student. Assessment of learner’s knowledge is mostly the only way how to determine suitable Learning Objects which should form individualised course.

1.2 Objectives of the Thesis

The major objectives of this master thesis are analysis of possibilities offered by some of the current popular online assessment tools and implementation of such a tool based on suitable open source technologies.

The first part describes some basics of the theory of assessment to introduce the reader into the problem and also contains a brief comparison of several web-based assessment systems which can be used to create and deliver online tests. At the end of the first part there is a description of ideal assessment system including the listing of all its major features.

The second part discusses technologies which were used to develop simple online assessment system called I-Test. It also further describes the design process and implementation of the application including its testing and possible feature extensibility.
2 The Theory of Assessment

This chapter introduces to the reader some basics of the theory of assessment and shows the importance of statistic analyses of test results in determining the quality of assessments.

Testing is one way to obtain reliable information about a certain subject although it is often difficult to assess complex subjects such as humans. Well formed assessments produce useful information which can be further analysed. Preparation of such tests requires many skills, lots of time and in the first place, evaluation of the quality of tests.

There are several major types of assessments which differ according to what properties they measure:

- *didactic tests* – to determine knowledge of students gained during learning
- *tests of abilities* – to determine abilities of a person in given area
- *psychological tests (personality)* – to determine personal properties of man

This thesis is focused on evaluation of students in e-learning and therefore only assessments of knowledge will be mentioned in detail. [1][2][3]

2.1 Types of Assessment Items

An assessment is formed by assessment items (questions). These items can be divided into two major groups: closed items and open items.

2.1.1 Closed Items

Closed items usually present the tested subject with set of choices. The subject is required to pick the most suitable ones. Advantages include high objectivity and the possibility of automatic scoring of this type of questions. On the other hand, there are several disadvantages. The subject can guess a correct answer. Furthermore, these assessment items do not test active knowledge but only the ability to recognise a correct answer.

2.1.1.1 Multiple-choice

Multiple-choice items are formed by a stem followed by a list of choices. The choices contain correct and incorrect answers called distractors. These items are the most common ones used in tests. Their scoring can be done automatically and can be further divided into questions with multiple responses or single response only. The number of choices in one question is
usually 3 or 4 but should not exceed 6. The probability of subject guessing the correct answer is defined as follows:

- for questions with single correct response:
  \[ P(N) = \frac{1}{N} \quad P(4) = 25\% \]
- for questions with multiple correct responses:
  \[ P(N) = \frac{1}{2^N} \quad P(4) = 6\% \]

2.1.1.2 True/False
The use of these items should be considered carefully. They are a special set of multiple-choice questions with single correct response and therefore the probability of guessing correct answer is \( P(2) = 50\% \) which is too high. However as mentioned in [2] if the number of items in a test is big enough, then such a test can be more reliable than a test formed by classic multiple-choice items.

2.1.1.3 Matching
These questions are formed by a stem and two sets of terms. The task of the subject is to pair the terms from both sets according to the rule which is contained within the stem. Matching questions retain all the benefits of multiple-choice questions (objectivity, easy automated scoring) and further decrease the probability of subject guessing the correct answer which can be defined as follows:

\[ P(N) = \frac{1}{N!} \quad P(4) = 4\% \quad P(5) = 0.8\% \]

2.1.1.4 Ordering
Ordering questions are similar to matching questions. The task is to order presented items according to the rule defined in stem. The probability of guessing the correct answer is again very low and equal to matching questions:

\[ P(N) = \frac{1}{N!} \quad P(4) = 4\% \quad P(5) = 0.8\% \]

2.1.2 Open Items
Open items require the subject to create the response. This eliminates the risk of guessing the correct answer. On the other hand, it decreases the objectivity of these items and also makes automatic scoring more difficult or impossible in some cases.
2.1.2.1 Short Text Entry
In this type of question subject is required to enter single word or short sentence as a response. Automatic scoring of these items is possible however brings some difficulties. The scoring systems must consider following issues:

- **Synonyms** – subject could express his response in different word with equal meaning.
- **Case of letters** – the case of letters should not be considered as important if it is not the target of the question.
- **Typing errors** – certain number of mistyped characters should be allowed in the response. Levenshtein distance algorithm can be used to determine the similarity between response and correct answer [4].

2.1.2.2 Essay/Composition
Questions of Essay type usually require subject to form more complex response. This is usually several paragraphs of writing. Automatic scoring of items of this type is impossible and therefore the objectivity of these questions is lower than in the case of closed items. This type of question can be sometimes replaced by the possibility to submit documents using file upload because the use of word processor is more comfortable and much safer than writing the text into text area of web browser.

2.2 Issues Related to Multiple-choice Items
Multiple-choice items are the most commonly used item type in online assessments. It is because of their easy automated scoring and objectivity. Preparation of questions of this type is not as trivial as it seems and there are several main problems related to this. [5]

2.2.1 Grammatical Cues
This issue is usually caused by incorrectly formed distractors which do not follow grammatically from the stem. Such distractors can be easily recognised and eliminated and the probability of subject guessing correct answer increases.

2.2.2 Long Correct Answer
Correct answer is usually more complete than other distractors and thus can be easily recognised. Therefore it is necessary to pay attention to the length of all answers and make sure that both correct answers and distractors contain similar number of characters.
2.2.3 Unintended Clue
The author of the question provides some clue he is not aware of. Subject can then eliminate incorrect choices and guess the correct answer. This issue relates to the use of same or similar words (synonyms) in stem and in choices. Also improbable distractors are hints increasing the probability of answering question correctly without having required knowledge.

2.2.4 Ambiguity
Ambiguity of a question is usually caused by unclear formulation of the stem or by incorrectly formed choices. This problem might occur in case that all of the choices can be correct in some case which is not specified enough in the stem.

2.3 Quality Factors of an Assessment
Following key factors are recognised and used to determine quality of an assessment: reliability, sensitivity, difficulty, objectivity and validity.

2.3.1 Reliability
Reliability of a test measures the influence of random factors on a test. The coefficient of reliability is a decimal number in range from 0 to 1. Good tests have reliability coefficients greater than 0.6 which guarantees that same test results can be achieved when testing is repeated several times.

\[ r = m(p - m)/(p \times s^2) \]  
(1)

Where:

- \( r \)........ reliability coefficient
- \( m \) ....... median of test scores
- \( p \) ....... number of items in a test
- \( s \)....... standard deviation of test scores, \( s^2 \)...variance of test scores

In case that the reliability coefficient is bellow 0.6 it can be improved by increasing the number of assessment items in a test or by removing the most difficult and easiest items. Multiplicative factor \( z \) can be calculated indicating how many times must be the number of assessment items increased to gain desired level of reliability. Following formula is also known as “Spearman-Brown prediction formula.”
\[
z = \frac{r_r(1-r_o)}{r_o(1-r_r)} \tag{2}
\]

Where:

- \(z\) ...... multiplicative factor
- \(r_r\) ...... required reliability
- \(r_o\) ...... obtained reliability

### 2.3.2 Sensitivity

The sensitivity indicates to what extent it is possible to differentiate among individual test results. The scores of subjects should cover the whole testing scale. The highest number of subjects should gain scores equal to arithmetic mean of all scores and the minimums should be at the ends of the scale close to maximum and minimum scores. Optimum value of this index should be greater than or equal to 0.25.

\[
i_t = \frac{s_N}{\bar{X}} \tag{3}
\]

Where:

- \(i_t\) ...... index of sensitivity
- \(s_N\) ...... standard deviation of test scores (N = number of test scores)
- \(\bar{X}\) .... arithmetic mean of test scores

\[
s_N = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{X})^2} \tag{4}
\]

### 2.3.3 Difficulty

Difficulty is represented by the index of difficulty. Optimum value is 50%. Values of this index higher than the threshold indicate that the test is too easy and vice versa lower values indicate that the test was too difficult. However in tests of basic knowledge, the desired value of the index should be much higher than 50%.

\[
I_D = 100 \cdot \frac{\bar{X}}{X_{\text{max}}} \tag{5}
\]

Where:

- \(I_D\) ...... index of difficulty
- \(\bar{X}\) .... arithmetic mean of test scores
- \(X_{\text{max}}\)..... the highest possible score achievable in the test
2.3.4 Objectivity
Objectivity can be defined as fairness of an assessment. The whole test must be constructed according to certain rules so that its results are objective. Exact scoring rules should be defined so that independent scorers assign the same score to the same test. Objectivity is mostly an issue of open type items. Therefore, it is necessary to determine how incomplete responses are marked. An objective test should also cover the assessed topic with respect to higher importance of certain parts and the number and difficulty of items should be in relation to this fact.

2.3.5 Validity
Validity of a test is one of the most important factors and determining this variable is difficult. There are four key criteria which affect validity of an assessment and should be considered when creating an assessment:

- assessment items should evenly cover the whole curriculum
- the difficulty of items should be adequate for the level of knowledge
- the type of test used must be suitable for testing particular phenomenon
- the assessed phenomenon should not be distorted by other influences
3 Analysis of Existing e-learning Systems

Due to the popularity and benefits of e-learning, there are many systems on the market nowadays. Platforms supporting e-learning can be either proprietary solutions developed to match all needs of a given institution or large software packages. Categorisation of e-learning systems is difficult and almost any tool does not fit exactly into any of the categories. However following types of e-learning systems are recognised [6]:

- **LMS** - is a suite of functionalities which simplifies the process of administering education and training. It is a complex system used by administrators, managers, instructors and learners. Its main purpose is to schedule, register, track and bill learners through courses. It allows learners to find suitable courses and register for them. System administrators and managers are provided with reports and statistics which help them manage and improve courses.

- **LCMS** – is a system enabling management, reuse and delivery of digital content which is usually stored in central repository of objects. LCMS is based on authoring tool which facilitates creation of learning objects and also their import from other formats. It is also capable of more detailed tracking of users than LMS.

- **Virtual university** – is a complete package of tools allowing delivery of instructor-led courses. Such systems combine all capabilities of LMS and LCMS and add features offered by collaboration tools allowing easy communication of the learners and instructors within the course and building of virtual classes and communities.

As stated earlier, assessment plays a very important role in e-learning and this was the main reason why some of the most popular e-learning systems were analysed to determine key requirements for an assessment system. If an e-learning system wants to succeed in very competitive market at a time when the boom of e-learning has slowed down, it must contain support for assessing learners.

E-learning systems can be divided into two major categories depending on the purchase costs, open source and commercial systems. The assessment features of several big Learning Management Systems as well as two web testing services for the purpose of covering broadly the problems of online assessments would be analysed. LMS Moodle, LCMS ATutor and LMS Ilias were chosen from the open source category because of their popularity. From many of the existing commercial systems, Microsoft Class Server would be evaluated mostly
because this system was chosen by the Czech Technical University in Prague as its e-learning platform. Analysis of commercial systems is often complicated because it is usually impossible or very difficult to get their evaluation version or some demo account. CTU readily provided an account allowing evaluation of all watched features offered by MCS.

The intention was to analyse possibilities of WebCT Campus edition which belongs to the category of Virtual universities and is the largest commercial e-learning system available. Unfortunately, WebCT does not provide any demo accounts for evaluation of their products and for the lack of information, it was not included in the survey.

The two analysed online assessment services are Questionmark Perception and Hosted Test. These services are dedicated to testing only and provide hosting of assessments on their servers including all software tools required to prepare these assessments. Both of the services are commercial and the price is usually calculated according to the number of tests taken by learners.

3.1 E-learning Standards

The boom of e-learning resulted in the development of many solutions which lacked consideration of compatibility of products of other competitors. As the rapid phase of e-learning has slowed down, the demands for migration of learning content from one system to another are becoming increasingly important. This led to the creation of several standards for e-learning. The most important standards and specifications in the area of e-learning and especially in online testing are IMS and SCORM.

3.1.1 IMS

IMS Global Learning Consortium was founded in 1997. IMS is concerned with developing specifications defining structure of learning resources to enable interoperability between learning content and learning management systems. The specifications consider both offline electronic resources delivered on CD-ROMs and other media and online resources distributed over the Internet. Its primary focus was on higher education but its output found use in wider learning contexts such as K-12 or corporate and government training. [7]

The most important specifications produced by the IMS Global Learning Consortium for e-learning and especially for assessment systems are the “IMS Content Packaging Information Model” and the “IMS Question and Test Interoperability” specifications.
3.1.1.1 IMS Content Packaging Information Model

This specification describes data structures representing learning content with respect to the exchange of the content among authoring systems, LMSs and runtime environments. The specification defines structure of a Package and Package Interchange File. IMS Packages are used in SCORM standard which will be discussed further in this text.

The IMS Package is a logical directory representing single units of reusable learning content. It contains physical files which form the learning material such as documents and multimedia files and also metadata describing bindings among the files and overall structure of the package.

Package Interchange File can be defined as transportation wrapper of an IMS Package and is used for its distribution mostly over the Internet. PIF is a single file compressed using ZIP algorithm and contains the whole package. Structure of the file is described on Fig. 3.1.

3.1.1.2 IMS Question and Test Interoperability

IMS Question and Test Interoperability specification defines abstract data model in UML for assessment items (questions), assessments (tests) and their results. It enables exchange of these objects between authoring tools, learning systems, assessment delivery systems, etc. Relations between these objects are displayed on Fig. 3.2. The latest IMS QTI specification at present is version 2.0. The specification defines XML description language called QTI XML which can be used to describe questions, tests and reports.
QTI recognises two types of assessment items: simple and composite. Simple items contain just one point of interaction. Composite interactions contain two or more interactions of the same type or different types of interactions. The interactions correspond to different types of assessment items. There are nineteen interactions specified in the standard. The interactions can be categorised into graphic and non-graphic interactions. Some of the interactions from both categories are almost the same. They differ only in the use of different means to express the task e.g. gap match interaction and graphic gap match interaction or order interaction and graphic order interaction. File upload is also one of the special non-graphic interactions which allows users to submit their own files to the server as a response. Together with the assessment items, the specification also defines ways how responses should be processed and scored.

The biggest benefits of this specification lie in the definition of common format for assessment items and response processing of some of the simple items. The specification is easily extensible to match future requirements.

Fig. 3.2 Participants of content exchange and their collaboration

3.1.2 SCORM

SCORM stands for “Sharable Content Object Reference Model”, which was developed by the U.S. Department of Defense. The steps which led to definition of this standard were
inefficiencies in training development and delivery across branches of DOD. It is a set of specifications and some of them were developed in prior decade by the AICC, IEEE, ARIADNE and IMS. The standard is focused on enabling immediate interoperability, accessibility and reusability of web-based learning content. It is based on broadly accepted technology standards like XML and JavaScript.

SCORM introduces Sharable Content Objects (SCOs) which represent atomic learning units comprising a SCORM course. The use of SCOs allows their reuse in more courses and also creation of highly personalized dynamic courses which are built “on the fly” to match specific needs of a learner. Courses formed by SCOs can be modified easily with lower costs.

The standard is still evolving and the latest version is 1.3.1 (also known as SCORM 2004). SCORM can be divided into four parts (books): SCORM Overview, Content Aggregation Model, Run-Time Environment, Sequencing and Navigation. [8]

3.1.2.1 Content Aggregation Model (CAM)
CAM book describes components of SCOs and their packaging to enable easy exchange from system to system. It defines requirements for building content aggregations like courses, lessons, etc. from basic objects. The packaging mechanism was taken from IMS Content Packaging Information Model which is described in section 3.1.1.1. All content packages are therefore provided with metadata which enhance search and discovery of the content.

3.1.2.2 The Run-Time Environment (RTE)
This book focuses mostly on communication between SCOs and LMSs. SCORM compliance guarantees interoperability across multiple LMSs regardless of authoring tools used to create the content. The RTE consists of following three main elements:

- **Launch** – defines relationship between LMS and SCORM content
- **API** – defines set of functions which are used to establish, maintain and close the communication between LMS and SCO
- **Data Model** – defines the set of data which can be set or retrieved from LMS by SCO

3.1.2.3 Sequencing and Navigation (SN)
Since SCORM 2004 the responsibility of determining sequencing of the content was directed to LMS. The content is sequenced according to set of learner or system initiated navigation events. LMS is responsible for creation of activity tree which represents the flow and
branching of the content and reflects learner’s interactions with the content and the strategy defined by the content author at time of design.

3.2 MS Class Server
MS Class Server is an e-learning solution developed by Microsoft. MCS can be categorised as LMS. This system is primarily designed for use in basic and secondary schools. With some modifications, it could be also used in universities. At the time of writing this thesis the latest version is Microsoft Class Server 4.0 but the version which is being used by CTU and was evaluated is MCS 3.0. [9][10]

MCS requires Windows 2000 Server or higher to be installed and uses SQL Server 2000 Desktop Engine as database which is shipped with the product. For large installations exceeding 1000 students it is necessary to obtain licence for SQL Server 2000.

Class Server was developed on the .NET platform and conforms most of the leading e-learning standards like IMS, SCORM and SIF. The whole system API is available to developers, which further simplifies integration of MCS within existing systems of given institution and also allows development of add-ins extending its functionality.

3.2.1 User roles in Microsoft Class Server

- *administrator* – is responsible for the operation of the system, manages teachers, students and classes
- *teacher* – creates learning materials, organizes students into groups, assigns materials to students
- *student* – studies learning materials and participates on tasks and assessments
- *parent* – can track progress of a student

One of the features which distinguish MCS from other online systems is that the user interface of a teacher is realised by a client application instead of traditional web-based access using browser. This breaks the limitations of HTML components usually used in user interface and simplifies creation of learning content. The other advantage provided by this solution is that the teacher can work offline and synchronise the work with the server when connected to the Internet.
3.2.2 Assessment Possibilities Offered by Microsoft Class Server

MCS does not contain stand-alone assessment modules like some other e-learning solutions. Assessments are represented by learning materials containing questions as special objects. Questions can be inserted from predefined templates using a wizard which simplifies the process. MCS supports following question types:

- multiple choice
- true/false
- gap fill
- short text answer
- matching
- long text answer - composition

The fact that questions are inserted directly into learning material brings some crucial problems. The first problem is in organisation of questions and their reusability in multiple assessments. MCS does not support question banks which could be used as question sources and because questions are not stored as separate learning objects they cannot be reused easily. The other problem is that MCS does not allow creation of randomised tests and cannot shuffle answers in multiple choice questions. Due to the character of assessments in MCS a test cannot have a specified duration; only the end time of the task can be set.

MCS is capable of generating reports of students’ results which can be exported to Microsoft Excel for further analysis. However detailed statistics are not presented.

Although it is possible to implement some of the missing important functionalities as it was done in case of CTU, the assessment features offered by MCS are poor in comparison with other systems. Good scalability and reliability offered by this product do not compensate high acquisition costs and its limitations.

3.3 MOODLE

Moodle [11] is an open source project distributed under GPL which was started by Martin Dougiamas in 1999. Since that time Moodle has by de facto become a standard in e-learning and is continuously developed by a team of programmers from all over the world. It currently has 70 language mutations. The latest version at the time of writing this thesis is 1.5.3 which has redesigned user interface to comply with WAI (W3C) criteria. Moodle is written in PHP scripting language and therefore can run on most modern operating systems. For data storage
Moodle best supports MySQL and PostgreSQL database systems but can work with any other database since it uses abstraction layer to access data from database. Installation and upgrades of Moodle are very simple and can be performed by less experienced users.

SCORM standard is also supported by one of Moodle’s modules and allows usage of SCORM content. However creation of SCORM complaint content is not supported.

Its modular structure allows for easy extensibility and the PHP language further facilitates the development. However as the project grows it starts facing the disadvantages of this scripting language and would benefit of use of Java or .NET which are more suitable for large scale applications.

### 3.3.1 Assessment Possibilities Offered by MOODLE

Due to modular character of Moodle assessments are realised by one of its modules. This system offers setting of various parameters of an assessment. The number of the parameters makes assessment module of Moodle unique because none of the evaluated systems offered similar functionalities. The key assessment options include:

- multiple attempts to take a test
- number of questions displayed per single page
- duration of a test
- shuffling of questions and answers
- displaying of results
- limit access to test from certain IP addresses
- password protect the test

Significant efforts were put into the reduction of the possibility to cheat and other unfair practices. Moodle offers special “secure” mode which requires JavaScript in order to run the test. Assessment is then displayed in full screen mode and disables some keyboard shortcuts, saving and printing of the test. However it is important to state that preventing users from obtaining a copy of the test during online testing is almost impossible and requires use of special client applications or dedicated computers.

Moodle offers a wide range of supported assessment items which are organised into categories. The system is also capable of importing and exporting questions to and from other popular systems. The most important import formats include: WebCT, Blackboard,
HotPotatoes and GIFT. Export formats offer the IMS QTI, GIFT and XHTML. In both cases, it is possible to use proprietary format of Moodle which can be used to share assessments among Moodle users.

Following assessment items are supported:

- multiple choice
- true/false
- short answer
- numerical
- calculated
- matching
- cloze (gap fill)

Moodle also supports adaptive questions. This type of questions is specific in the way that the question is modified every time the user submits it. In the simplest case the question just provides the user with different feedback until the student reaches correct answer. A penalty is being calculated considering the number of attempts required to answer the question correctly.

The assessment module of this system cannot generate any test statistics. Also it is not possible to export the results for further analysis. This functionality is however under development and probably appears soon in future versions. Despite these minor drawbacks Moodle can be considered as a very good system for creation and delivery of assessments.

3.4 ATutor

ATutor [12] is open source web-based LCMS system. This system was designed with the focus on accessibility and adaptability. It complies with WCAG 1.0 accessibility specifications at the AA+ level. A text-to-speech system is involved in reading labels of links to people with sight disabilities. ATutor complies with SCORM 1.2 standard. It is capable of importing SCORM packages and also exporting the content created using its authoring module in SCORM compatible format. The support of this standard is at a higher level than in LMS Moodle.

This system was developed in PHP scripting language and uses MySQL database to store its data. Installation of this system is simple and usually any problem-free.
Although the user interface conforms to strict specifications, its quality is not as good as in case of Moodle and the overall design and layout is not so clear and well-arranged.

3.4.1 Assessment Possibilities Offered by ATutor

ATutor is using separate assessment module to deal with tests and surveys. Test options which can be specified are not as many as in case of Moodle. Author of a test can specify start time, end time and number of attempts. Questions used in a test can be randomly chosen from predefined question banks and the total number of questions in a test can be set. Integration of tests and surveys into single module seems as a good alternative. In case of a survey all responses are anonymous. LMS Moodle uses separate modules for tests and surveys.

Supported assessment items are:

- multiple choice
- true/false
- short text answer
- composition

Evaluation of the assessment module discovered its incapability to score short text answers automatically which can be limiting. Also it is not possible to create multiple choice questions with multiple correct answers.

3.5 Ilias

Ilias [13] is another open source LMS founded in Germany in 2000. Currently it is being developed by a network of universities with centre at the University of Cologne. Ilias has many installations all over the world but most of the installations are in Germany. Ilias was the first free LMS which conformed to SCORM 1.2 Level LMS-RTE3 and therefore guaranteed platform independent reuse of content. It is localised into many languages. However, the number of supported languages can hardly be compared with LMS Moodle and also Czech localisation was incomplete.

Ilias is developed in PHP scripting language and uses MySQL database to store its data. The installation of the system was not as easy as in case of previous two open source systems and some difficulties occurred.
3.5.1 Assessment Possibilities Offered by Ilias

Ilias has modular structure and assessments are maintained by a separate module. The module was created due to wide support of e-learning standards capable of importing and exporting tests and questions packaged according to IMS QTI specification. Ilias can generate simple test statistics which can be exported in CSV format for further analysis. According to rules specified by test author, the final test score can be converted to a grade. The assessment engine can only display a single question per page. On the other hand, it is possible to set time limit for each question.

Supported assessment items:

- multiple choice questions (single response, multiple responses)
- cloze question (fill in gaps)
- matching
- ordering of texts and images
- image map
- short text answer
- essay / composition
- java applet

Ilias offers wide range of supported question types but the user interface was not very friendly. Also, no open type questions can be scored automatically and manual scoring was very uncomfortable.

3.6 Questionmark Perception

Perception [14] is an assessment management system which enables authoring, scheduling, delivery, and reporting on surveys, quizzes, tests and exams. This system comprises of several standalone tools including authoring manager, server software and other utilities simplifying creation and delivery of tests. Questionmark also offers hosting of assessments so there is no need to buy cost expensive hardware and the server software package. The server software uses Microsoft SQL or Oracle database to store its data and can be easily integrated within other information systems thanks to existing API. Perception is compliant with SCORM 1.2 standard.
Authoring of questions and tests is performed using desktop client application. Questions are organised into question banks by learning objective.

Assessments can be delivered via web browser, PDA, CD-ROM or paper. High-stakes exams can be delivered over the Internet using special secure client application emulating standard web browser. Users are prevented from saving, printing or capturing the content of the test. Among many of the options of an assessment are randomisation of questions and answers and also adaptive branching which allows dynamic modification of an assessment based on how questions are answered. Reporting of test results can be made using several predefined reports that can be further customised. Perception also generates statistics on assessment items.

Questions in assessment can be delivered all on a single page or on question per page basis. Perception supports the widest range of assessment items from all evaluated systems:

- true/false
- multiple choice questions (also with multiple responses)
- matrix question
- matching
- cloze (gap fill)
- ordering
- short text answer
- numerical
- essay / composition
- spoken response question
- drag & drop
- image hotspot
- flash question

3.7 Hosted Test

Hosted Test [15] is an online assessment hosting service. The service is based on online software allowing authoring and delivery of assessments. Hosted Test is built using open standards with Microsoft Windows, IIS, .NET and SQL Server technology. Appearance of all assessments created using this software can have fully customizable look and feel according to the needs of the institution. Pricing of this service is based on a number of tests taken so that there would not be any additional fixed costs. This service has wide capabilities of
importing and exporting of tests, questions, student data and reports. Unfortunately the support of e-learning standards was not mentioned on the website of the product. Microsoft Excel can be used as data source for students’ data and also for further analysis of reports which can be exported in this format.

Many parameters of assessments can be set to satisfy demands of wide range of users. Complex and large assessments can be divided into several chapters. It is possible to specify time limits for each chapter or for the whole test and also to state whether test must be finished within single session or if multiple sessions are allowed for test completion. Hosted Test supports templates of tests to speed up creation of new assessments according to the ones previously created. It is also possible to combine several templates together to form a new test.

Hosted Test also introduces personalised email invites and notifications to inform users that the new assessment has been published and also to notify them that they have not yet completed the required assessment.

Question types:

- multiple choice (with single response and multiple responses)
- essay / composition
- matching
- short text answer

Additional question types can be implemented according to the needs of given user. Some special assessment features include “conditional question branches” which allows creation of adaptive tests that modify their structure according to the responses of subject.

3.8 Outcomes of the Analysis

Analysis of the systems mentioned above provided key requirements for a good online assessment tool which could be used to author and deliver tests. None of the open source solutions contains ideal assessment module but combination of all these systems could provide the best platform for preparation of assessments. Unfortunately the analysis of the two online services mentioned in 3.6 and 3.7 was not done fully because only the information from the website of these products were used and available sample tests taken. On the other hand, all of the open source systems were evaluated thoroughly including installation on local
system and careful testing of all advertised features. Direct evaluation discovered in some cases problems in user interface which was not very user friendly.

3.8.1 **Key Features of an Assessment System**
Following section tries to summarise the most important features of an assessment system.

A bullet point list of the key features follows:

- *security* – is crucial for every assessment system and must be considered accordingly
- *conformance with e-learning standards* – this allows integration with existing systems and exchange of learning content created in different authoring systems
- *customisable look & feel* – there should be some possibility to customize the layout of the application according to the needs of given institution
- *reporting* – the system should be capable of generating various reports and statistics of assessments and questions to provide feedback to the author
- *import & export* – importing and exporting features from formats used by systems other vendors is important for every new application
- *accessibility* – the system should be made accessible by all users and various disabilities should be considered
- *internationalisation* – the system must support easy internationalisation

3.8.2 **Key Attributes of an Online Assessment**
Online assessments should allow the author to set various properties to limit access of learners to the assessment. Typical constraints include start time, end time and duration. Questions included in an assessment should be chosen from question banks and the possibility to randomly select specified number of questions from a bank should be supported. Test author should have the possibility to allow learners to see scored test with correct answers and also to let unsuccessful students to take the test again (multiple attempts). Questions in test could be displayed all on single page or each question on separate page to enable use of adaptive questions. Automatic or manual saving of work in case that all questions are displayed on single page should be implemented to prevent users from loosing their work. Also the possibility to restrict access to the test from certain hosts is sometimes required.
3.8.3 Supported Assessment Items

The more types of assessment items the system supports the better. Authors of tests can then use more suitable question types to test certain aspects and increase overall validity of a test. The ideal list of supported question types is described in IMS QTI specification. The most important assessment items are as follows:

- **multiple choice questions** – including both types with single response and multiple responses, shuffling of choices
- **true / false**
- **short text answer** – this type of question should be scored automatically considering synonyms, typing errors, etc.
- **essay**
- **matching**
- **ordering**
- **gap fill (cloze)**
- **image hotspot**
- **file upload** – to allow users to submit larger documents

All the questions should allow the author to specify feedbacks that are displayed after completion of the assessment or also hints which are displayed dynamically during the test to help user to answer the question. Hints can usually find their place in tests whose main purpose is the promotion of learning.

3.8.4 Summary

In the conclusion of the analysis it is necessary to say that assessment system is very complex application with specific requirements. Every such application must pay attention to security issues, internationalisation support, good user interface, accessibility and extensibility.

Development of such application would go far beyond the scope of master thesis and is mostly a task for a team of developers and not for a single person. Therefore it was decided to design and implement only a very simple system with respect to security, accessibility, good user interface and its future extensibility.

The outputs of this analysis were considered throughout the design of I-Test and allowed to focus on the most problematic parts.
4 Design

The first step during the design process was the selection of the architecture of the system. This step was predetermined by the assignment of this thesis and the client-server model was therefore chosen. All of the analysed systems had the same architecture which is the most common one in networked environments like intranet or on the Internet. Vast majority of the analysed systems uses standard web browser as the client side of the application. This step has many benefits but also some drawbacks. Almost all computers are equipped with some sort of web browser today which makes the application available to wide range of users without the need to install any software into their computers. The use of standard web browser is cost effective solution because it eliminates development and maintenance of separate pieces of software. But on the other hand the use of different browsers brings some possible incompatibility issues and the biggest advantage of standalone client applications is in much more comfortable user interface. However the use of Flash, DHTML and JavaScript can slightly diminish this major benefit.

4.1 Programming Language

The other important decision was the choice of programming language which was used to code the server part of the application. PHP scripting language is probably the most popular language for programming of web applications and also all the analysed open source e-learning systems were developed in this language. The ease of development and wide range of supported operating systems make PHP an attractive candidate. Microsoft .NET was not considered as a suitable technology because of its very close bindings to commercial products and more difficult deployment. The winning candidate is Java [16] with its Servlet and JSPs technologies [17]. The key factors supporting this decision are the following:

- platform independence
- multi-threading
- higher security
- better performance than interpreted scripting languages
- more suitable for development of large projects
- strong type safety
4.2 Internationalisation

Internationalisation (sometimes also referred to as i18n) is very important for every application which is used by people speaking different languages. This fact is much more significant for web-based applications on the Internet that are accessed by people from all over the world. Although English has become very popular and widespread mostly because of the Internet, working with the application in one’s native language gives the user more confidence and familiarity.

Internationalisation of the application is a step which should be considered at the initial stages of the development to avoid possible difficulties of implementing this feature later. The use of Java is apparently a very good choice in this respect because this language supports natively UNICODE encoding. This encoding is virtually the only encoding allowing localisation of the application to all worldwide languages including Chinese, Japanese, etc.

Java furthermore supports internationalisation process by its “Resource Bundles”. Resource bundles have been introduced since Java 1.4 and are commonly used to create and maintain more language mutations of an application. These structures are used to store locale-specific content like labels or text messages. Resource bundles can be also used in traditional desktop applications. They are represented by simple text files which have following structure:

key1=text1
heading.key2=text2
label.key3=text2

The locale of the file is determined according to the structure of its filename. The filename consist of the name of the bundle, underscore and code of given language according to ISO-639 followed by the extension .properties. Optionally, to differentiate between countries speaking the same language the country code according to ISO-3166 preceded by underscore can be appended after the language code. Examples of the naming conventions follow:

Resource1_cs.properties
Resource1_en_US.properties

The best character set to be used to encode characters of localised content is of course UNICODE. Localised texts can then be simply retrieved in the application according to their key by calling some API functions from the library of resource bundles.

The biggest advantage of resource bundles is that new language mutations can be created without the need to recompile any classes. In addition, the user’s locale can be determined
from the operating system by client application or is sent by web browser in every request therefore determining the correct locale can be done automatically and the whole process is transparent to the user.

4.2.1 Encodings Used on Web Pages
Almost everybody working with the Internet and email must have experienced some problems with character encodings. Several encodings exist only for interpreting Czech characters on the Internet (Windows 1250, ISO-8859-2, etc.) and similar situation exists with other non English languages. Problems with encodings have mostly historical reasons and should not be an issue with current modern operating systems and browsers which support a variety of encodings. The UTF8 encoding was selected as the only and universal encoding supported by I-Test. Together with UTF16 is considered as default encoding for XML files.

UTF8 is variable length character encoding for Unicode. It uses one to four bytes to code single character and is backward compatible with ASCII. It is loosely based on Huffman coding and the most frequently used characters have the shortest codes. This means that they are encoded on less bytes than the rare characters. UTF8 is widely supported by the IETF which requires all Internet protocols to recognise at least this encoding and the IMC recommends that all email clients should support this encoding. The use of UTF8 unfortunately brings also some drawbacks. The most significant issue is increased size of web pages using other than non ASCII characters with UTF8 because legacy encodings require only one byte to encode a character. The variable length of UTF8 makes the algorithms processing the input more complicated and less powerful. However all these minor drawbacks outweigh difficulties connected with maintaining more legacy encodings.

4.2.2 Editors of Resources
As was stated in previous section, resource bundles are files in simple text format. Maintaining several language mutations of an application can be difficult task without the use of any tool designated to this purpose. There are many editors of resource bundles which can be used to easily maintain large numbers of files containing localised resources. These editors usually display selected language mutations to facilitate their translation and are capable of performing operations like renaming of the keys on all resource files at one time. Some of the editors also offer the possibility of parsing source files for text messages and modifying sources to support localised content. However this approach is not recommended and localisation should always be considered at the early stages of design process. Sample screen
of the editor which was used to maintain localised resources of I-Test can be found on Fig. 4.1.

4.3 Security

Security of web applications on the Internet is a big issue nowadays because of the ubiquitous presence of hackers. Therefore security of the application is considered as major aim during the whole development process. The security issues can be divided into two main categories:

- anonymous users trying to gain access to the system, modify or corrupt data
- authenticated users of the system trying to access unauthorised parts of the system

Both of these categories represent serious security threats which must be prevented. Anonymous access to the system was disabled and only users with valid accounts can login to the system. Following sub chapters contain list of the most common security threats and state the protection of I-Test against them.

4.3.1 Manipulation of URL

Users who work with I-Test in correct manner only visit pages that are accessible using links on the web pages in their browser. It is also possible to enter an URL directly into address bar
and thus access web pages that are under normal circumstances hidden to the user. Another approach is to modify parameters passed in the URL. Both of these techniques could lead to unauthorised access to various resources. In most cases users with lower privileges would try to access pages requiring higher security level. To prevent this type of disallowed access to the system every request to a page or action is processed by servlet filter which verifies whether the given user has sufficient privileges to access the resource. In case of insufficient privileges or unauthenticated user the access is denied and a record notifying about the attempt is logged. The system contains three security roles corresponding to the three user roles of the system: administrator, test manager and student. Members of given roles are only allowed to access pages within their security role. This for example prevents students from executing any administrators’ or test managers’ actions but also administrators to execute actions of test managers.

4.3.2 Use of Secure HTTP Protocol
Due to the insecure character of the Internet and the fact that the application is working with highly confidential data, HTTPS protocol was chosen for the purpose of securing the communication channel between server and clients.

The use of encrypted communication prevents unauthorised users from obtaining confidential data like usernames and passwords using network traffic analysers (packet sniffers). The biggest drawbacks of this approach are in reduction of the performance because every response of the server must be encrypted and client’s requests decrypted. To decrease this performance penalty some systems use secure connection only during the login process when user names and passwords are being sent. It is believed that this approach could be insufficient and therefore the use of encrypted communication is recommended for all parts of the application. Due to the fact that I-Test is still in the state of prototype secure communication was not compulsory but was used during the testing. The installation manual describes how this can be changed.

4.3.3 SQL Injection Attacks
This type of attack is caused by incorrect handling of user input in the application. Severity of this exploit is very high and can allow the attacker to gain control over the whole system by executing his own SQL queries injected through parameters passed directly from clients without performing any additional checks. Typical example of this attack is when user enters specially altered text into input field.
Typical SQL query which validates login of a user looks like this:

```
SELECT * FROM users WHERE user_name = '$user_name' AND password = '$password'
```

In case that the user provides following string as password:

```
anything' OR 'x'='x
```

and this input is inserted directly into the query. The query then returns always true and access to the system is granted to the user without knowing the correct password. Therefore it is necessary to check the input of user and escape all possible “dangerous” characters.

The given example is very simple but unfortunately very efficient. Due to the use of persistency layer and prepared SQL queries, the risk of this type of attack was completely eliminated.

### 4.3.4 Brute Force Attacks

This type of attack uses brute computing power to reveal the user’s password by trying all possible combinations of characters. This type of attack can be combined together with SQL injection to avoid its detection. These attacks are usually used to gain access to the system. In case of I-Test the risk of successful attack was minimised by adding a delay of several seconds and logging invalid login attempts when incorrect password was entered. Users of I-Test are not allowed to use passwords shorter than five characters to make brute force attacking more difficult. This step cannot prevent attackers from succeeding but can provide system administrators with sufficient time to recognise this type of attack and perform further actions.

### 4.3.5 Dictionary Based Attacks

The efficiency of brute force attacks is very low and it usually takes significant amounts of time to reveal the password and gain access to the system. Such attacks are soon discovered and blocked before the password can be guessed. Therefore it is important to mention another form of password guessing method. This type of attack is based on the fact that most people use weak passwords which are based on meaningful words or names. This technique uses dictionaries of most commonly used passwords to guess the correct one and can succeed much faster than the brute force attack method. The only way how to prevent this method is to verify the passwords entered by users and not to allow the use of such weak passwords. However this technique was not implemented in the application.
4.3.6 Protection of Assessments

Protection of delivered assessments belongs also to the chapter dedicated to security of the system. Distribution of any materials using standard web browser over the Internet has the drawback that the author loses control over it. In case of online assessments it might be desired to prevent user from obtaining a copy. There are several techniques which can be used to make this more difficult to less experienced users. However, it is not possible to avoid this issue completely without the use of special client applications or dedicated computers. The use of special client applications prevents user from obtaining a copy of the test by saving, printing or capturing the screen but this approach cannot stop users from taking a picture using a camera. With the fast development of mobile phones which are equipped with high resolution cameras, this is becoming a real problem. To avoid this risk, assessments must be taken in specialised assessment centres where the users are monitored permanently either by invigilators or video surveillance cameras. Such centres are equipped with computers dedicated to this purpose not allowing execution of anything else than one instance of web browser to access the assessment. The use of assessment centres is very expensive and therefore is usually used only for high-stakes exams. [19]

4.4 Model-View-Controller

Current programming techniques focus on producing maximally reusable code to facilitate development of other related projects. To follow these trends, several design patterns were implemented to increase reusability of the code and also extensibility of the application which was one of the main goals. Most of the web applications especially those written in scripting languages for web such as PHP combine business logic with user interface and control logic. For small projects this approach saves some time during the development but it creates a code which is not reusable and modification to one of the layers requires changes to the others. This simple approach brings big problems for large applications consisting of many pages and therefore layered architecture known as Model-View-Controller is being used. Sometimes this architecture is also referred to as Model 2. Its predecessor Model 1 did not separate business logic (Model) from presentation layer (View). The MVC architecture was first used in Smalltalk and gave birth to many frameworks. The architecture defines following three layers:

- **Model** – represents the business logic of the application, usually formed by Java objects, JavaBeans or in enterprise applications by EJBs.
- 56 -

- **View** – represents presentation layer of the application to display its **model**. It typically consists of HTML pages to display static content and JSP pages to generate dynamic content.

- **Controller** – is generally represented by Java servlets and is responsible for accepting and processing requests of clients.

The interactions between the components forming the MVC are displayed on Fig. 4.2.

![MVC Architecture Diagram](image)

**Fig. 4.2 MVC architecture**

The client’s requests are processed by a controller which invokes appropriate actions on model of the application. The request is then dispatched to the view which is responsible for presentation of the current state of the model of the application. [20][21]

### 4.5 Jakarta Struts Framework

Starting development of an application from scratch is a very difficult, time consuming and in most cases somebody has already developed something similar. This led to the development of many specialised frameworks whose main purpose is to facilitate and speed up the development of applications and especially web based applications which have many common features. A framework is a reusable, semi-complete application that can be specialized to produce custom application. Application frameworks form common ground to provide a reusable structure that can serve as the foundation for more specialised product. The structure of an application built using a framework is displayed on Fig. 4.3. Jakarta Struts is a
framework implementing the MVC architecture. There are many frameworks providing similar functionality like Turbine and Velocity. The Struts framework was chosen mostly because of excellent stability, good support by many programming environments and lots of available literature and documentation. [20][21][22]

<table>
<thead>
<tr>
<th>Extension 1</th>
<th>Extension 2</th>
<th>Extension 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framework</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 4.3 Difference between the use of Framework and Library**

### 4.6 Database

The use of good database system is an important performance factor of every application. Most of the database systems used nowadays are traditional relational databases based on relational model supported by strong mathematical theory. Almost all modern programming languages are object oriented. Objects are used to model real world and to reduce its complexity by decomposing it into smaller pieces. The use of relational databases becomes cumbersome with increasing complexity of objects and this brought place for modern object databases. Although modern object oriented databases remove the drawbacks of traditional relational databases and in some aspects also provide better performance, their use is not so popular mostly because of their lack of formal mathematical foundations and this brings weaknesses in their query support. Database systems can be divided into following categories:

- **RDBMS** – Relational Data Base Management System
- **ODBMS** – Object Data Base Management System
- **ORDBMS** – Object Relational Data Base Management System (hybrid systems)

Traditional relational database was chosen for use but because of the object oriented design of I-Test, an object-relational mapping technique was used to facilitate the development. This will be described in detail later in chapter 4.7.

Several databases were considered for use in I-Test to store data. Oracle was initially intended for use of the database for development and release. However, since commercial use of this database is expensive and hardware requirements of this product are very high thus requiring use of stand-alone database server, lightweight open source database system was used instead.
4.6.1 MySQL
One of the possible candidates was MySQL database [23], the most popular open source database which is frequently used together with PHP scripting language and Apache web server. All the open source e-learning systems mentioned in this text supported this database. The reasons of wide use of MySQL are its performance, reliability and easy administration. Furthermore, it is available for all major operating systems. However, the high performance is paid for with limited features. MySQL version 4.0 does not support sub queries, views, stored procedures, triggers and the support for constraints is also very limited. The latest version of MySQL is version 5 which was released for production on October 24, 2005 removes these drawbacks and could further increase the number of installations of this database. At design time of I-Test, the version 5.0 is not available yet. Due to the drawbacks mentioned above, version 4.0 was not used for development.

4.6.2 PostgreSQL
PostgreSQL was another open source database considered. Its development started in 1986 at the University of California at Berkeley.[24] Since its beginnings, it evolved into a powerful and reliable database. It is available for all major operating systems including Linux, Unix and Windows. Unlike MySQL database, PostgreSQL is a full featured database engine supporting foreign keys, views, joins, stored procedures, triggers, etc. It also includes most SQL 92 and SQL 99 data types. Compared to MySQL 4.0, the biggest advantages are support of sub queries, the possibility to define different character encoding for each database and easy backups. The latest version of PostgreSQL at time of writing this thesis is 8.1. All the benefits mentioned above and other less important details led to the decision to use PostgreSQL as a primary database for data storage and retrieval.

4.7 Persistency of Java Objects
As mentioned above, most of the available database systems are traditional relational databases with their benefits and drawbacks. Database access in Java is unified because of JDBC API which defines functions used to connect to the database and perform SQL queries. But the use of JDBC API together with object oriented approach requires many unnecessary routines which only serve to two purposes: to load current state of the object and to store all its modifications back to the database. This approach is very troublesome and requires changes to the code when the database structure is modified. Luckily this exchange of
information can be maintained by object-relational mapping technology (ORM) which is responsible for persisting Java objects using relational database.

There are more than sixty object relational mapping products [25]. Some of them are commercial but these were not considered due to the character of the application which is based on open source technologies. The choice of the framework can be a limiting factor of the application and its change usually requires significant modifications of source codes.

To allow different persistency framework to be used or possible future migration to EJBs which are usually closely bound to specialised high performance application servers, simplified DAO design pattern was employed to separate all methods accessing data to dedicated classes which can be later easily modified with minimum impact on the rest of the application.

From the many existing frameworks, two of them were considered to be used in I-Test: Hibernate 3.1 [26] and the Apache OJB 1.0 [27]. Both frameworks provide similar functionality but Hibernate was chosen because of its wide usage, many articles, tutorials, examples and well documented API that were available for this system.

### 4.7.1 Hibernate

Hibernate is an open source and free technology enabling persistence of Java objects using ORM. Hibernate not only enables mapping of Java classes to SQL tables and Java data types to their SQL equivalents, it also provides a complex object query language called HQL.

Persistency of objects using Hibernate is transparent and there is no need to inherit all the objects from common parent or to implement some interfaces which would take care of maintaining the state of the object. Some other frameworks use byte code manipulation which is performed during the build time of the application to achieve this however this is not case of Hibernate. It uses Java reflection to enable transparent object persistence. [26][28]

Hibernate pushes the unified database access in Java using JDBC further and supports many database dialects for all major SQL database systems including MySQL, PostgreSQL, Oracle, DB2, Sybase, Microsoft SQL Server 2000, etc. It translates HQL queries efficiently into native SQL conforming to selected database. Furthermore Hibernate is capable of caching queries and because of this feature, it can increase the performance of the application.
The key components of Hibernate are the XML configuration files containing appropriate mappings. Hibernate is then able to create the whole database structure from these mappings and also to generate POJOs which represent classes corresponding to SQL tables.

4.8 Accessibility

Accessibility of the system was also one of the key factors considered during the design stage. Maximum efforts have been put into the design of user interface to make it as simple as possible but retain maximum clearance and accessibility for users with disabilities. Web based applications use the HTTP protocol to communicate with client browsers. The interaction is stateless and consists of a sequence of requests and responses. Web server responds to client request by sending a web page which is a document usually coded in the HTML or XHTML format.

4.8.1 HTML and XHTML

HTML is a markup language which is used to create web pages. The first versions of HTML had loose syntactic rules to allow easy creation of web content by less experienced users to popularise the web and browsers tried to guess the intended layout of the document and interpret the incorrectly formed documents. The specification of HTML is currently maintained by W3C consortium. Because of the loose syntactic rules and many cross-browser related problems with rendering of the content the consortium created a stricter standard based on XML language called XHTML. XHTML is a parallel branch to HTML and is currently being the recommended markup for creation of web pages due to its strict structure. Unlike traditional HTML format, XHTML separates formatting and content of a document by using Cascading Style Sheets to define layout of the document. [29]

Because of the benefits mentioned above, XHTML 1.0 transitional which is widely supported by all modern browsers was used. Valid documents are therefore rendered in a very similar way. Every webpage should be accessible regardless of what browser is being used and this goal was achieved.

4.8.2 Cascading Style Sheets

CSS is a technology which facilitates definition of the presentation of documents written in HTML or XHTML. CSS also helps to maintain the same look and feel over the whole web site by allowing external declaration of formatting rules. The use of CSS was inevitable because of the choice of XHTML and also brought benefits in simply customisable layout of
the pages. Only by modifying the single style sheet can all the colours, font-sizes and placement of some of the components be changed.

A lot of effort was put into the design of accessible user interface using CSS. The contrast colours of texts and links were chosen according to strict specifications requiring defined colour brightness difference and colour difference of foreground texts and their background. Colour brightness is determined by following formula:

\[
Brightness = ((R \times 299) + (G \times 587) + (B \times 114))/1000
\]  

(6)

Colour difference is defined as:

\[
C_{diff} = \max(R_1, R_2) - \min(R_1, R_2) + \max(G_1, G_2) - \\
\min(G_1, G_2) + \max(B_1, B_2) - \min(B_1, B_2)
\]

(7)

The recommended differences are 125 for colour brightness difference and 500 for colour difference. [30]

Another key factor which is luckily becoming standard on most modern web sites was relative definition of font size allowing user to increase or decrease font size to match his needs. This feature is especially useful for people with sight disabilities.

The application is using one style sheet for displays and another one which is used for printed documents. Special style sheet is also used to display assessments by students disabling printing of the assessment.

4.9 Templating

Sometimes the formatting features of CSS are not sufficient and changes to the structure of XHTML pages are necessary. Without the use of any templates which define the layout of the application, this task can be complicated because it requires modification of all the pages that form the presentation layer of the application. Luckily, Struts framework provides powerful templating mechanism which allows common definition of a layout. The part of the framework responsible for doing this is called “Tiles”. Its name is almost self-explanatory. The whole page is divided into functional tiles which are then assembled together using rules contained within global templates.

The template used in the case of I-Test defines a very simple layout which comprises of four parts: header, menu, content and footer.
Every web-based application is using some kind of web server which is used to host the application. There are many commercial and open source web servers available on the market nowadays. The requirement on the server because of the use of Java Servlets and JSP pages was that it must provide servlet container conforming to Java Servlet and Java Server Pages specifications developed by Sun Microsystems. One of the most popular solutions is Apache Tomcat [31]. Tomcat is an open source project developed by Apache Software Foundation in Java programming language and therefore can run on any operating system that has an implementation of JVM. To develop and debug I-Test the Apache Tomcat version 5.5.9 was used.

**4.10 Java Servlet Container Tomcat**

Every web-based application is using some kind of web server which is used to host the application. There are many commercial and open source web servers available on the market nowadays. The requirement on the server because of the use of Java Servlets and JSP pages was that it must provide servlet container conforming to Java Servlet and Java Server Pages specifications developed by Sun Microsystems. One of the most popular solutions is Apache Tomcat [31]. Tomcat is an open source project developed by Apache Software Foundation in Java programming language and therefore can run on any operating system that has an implementation of JVM. To develop and debug I-Test the Apache Tomcat version 5.5.9 was used.
5 Implementation

I-Test was implemented in JDeveloper 10 [32] a free development environment by Oracle. JDeveloper was not released as open source but is provided at no charge and is being continuously developed. It is a competitive IDE to another popular product Eclipse by IBM [33]. JDeveloper is a suitable platform for development of web services and J2EE applications in Java and was chosen mainly because of its good support of Struts framework. It provides visual editor for configuring Struts controller allowing easy creation of page flow. Example diagram representing structure of the application which is created using the editor is displayed on Fig. 5.1. Unfortunately the overall page flow diagram of I-Test cannot be presented due to its dimensions.

Fig. 5.1 Part of simplified page flow diagram

The page flow diagram consists of JSP pages, actions, forwards and links. JSP pages represent the presentation layer of MVC design pattern and are formed by combination of traditional
HTML tags, tags from Strut’s tag libraries and in rare cases when the functionality offered by powerful tag libraries was not sufficient also by scriptlets. Actions in Struts are always represented by Java Servlets which are responsible for calling methods of business logic and forwarding the view to appropriate JSP page. Forwards together with links form the structure of the application and define the data flow. Forwards are used by the controller to dispatch to view to appropriate JSP page. Links represent traditional HTML links which allow user to navigate through the whole application. It is necessary to state that even the partial diagram on Fig. 5.1 does not contain all links and only the most important ones were presented.

5.1 Context Diagram

Context diagram of the application is quite simple. The purpose of this diagram is to determine the boundaries of the system and its potential users.

I-Test identifies following three user roles:

- Administrator
- Test Manager
- Student
More detailed specifications of actions available to the users of the system are presented on use case diagrams in following section.

5.2 Use Cases
Following diagrams represent the actions which can be performed by its users. The action manage usually includes operations like create, edit and delete. These operations were grouped together to simplify the diagram.

Use case diagrams were split into three parts according to user roles because each user has almost unique operations. Only the operation allowing users to change their settings and update user details are common to all of the users. In the case of Test Manager and Student, the operation to view the scored test is the same too.

5.2.1 Administrator
Administrator of the system is responsible for creation, deletion and editing of Test Managers and can also view and delete the access log of the system.

![Fig. 5.3 Use case diagram of Administrator](image)

5.2.2 Test Manager
Test Manager is the most complex role of the system. This role is a combination of teacher and content author. Test Manager has more responsibilities than in other systems where the user management is usually governed by administrators of the system. The control over
students was assigned to Test Manager to provide this user with higher independence and more flexibility. Test Manager can also import students to the system from CSV files and organise them into groups. The main tasks of Test Manager are however the creation of question banks, questions and tests. Prepared tests can be then delivered to students or student groups. After delivering the assessments and their completion by students Test Manager can display the results and statistics.

**Fig. 5.4 Use case diagram of Test Manager**

### 5.2.3 Student
Student is the simplest user of the system. Among the actions available to a Student belong only: view list of available assessments, take an assessment and display its results.
5.3 Structure of the Application

The source codes were organised into several Java packages to separate parts providing similar functionalities. The main package is called `itest` and within its root contains mostly only the classes described in following chapter. This main package is further divided into several sub packages:

- **actions** – contains common actions such as login or logout and is further divided to:
  - **admin** – contains actions available to an administrator
  - **manager** – contains actions available to a test manager
  - **student** – contains actions available to a student
- **dao** – contains classes that are used to access the database
- **dto** – contains classes used to interact with presentation layer
- **exceptions** – contains exceptions thrown by the application
- **forms** – contains classes used to pass data from and to HTML forms
- **util** – contains miscellaneous utility classes

5.4 Class Diagram

This section presents the class diagram (displayed on Fig. 3.1) of the main classes forming the business logic of the application in UML. All of the classes in this diagram are direct descendants of `GenericElement` which implements some methods used to compare the equality of two instances and calculate hash code of an object. The equality of two instances is based on the type of the object and on the value of its `id` property which serves as primary
key in SQL representation. To maintain readability of the diagram this inheritance was not displayed. The classes displayed on the diagram only contain their private properties because the majority of their public methods are only setters and getters of these properties.

Fig. 5.6 Class Diagram of I-Test in UML
5.4.1 Description of Classes

This section provides a brief description of classes mentioned above and their properties. Most of the classes are rather simple and need to be extended in future. A table with properties including their mapping to SQL representation is only attached to more complex classes. In case that mapping to SQL is missing then the property is not stored as a field in the database and usually represents some container which is populated by the persistency framework when it is being accessed.

All the classes are using wrapper classes of standard Java types because of the persistency framework. The type Set which is used as generic container corresponds to `java.util.Set`. More specialised implementations of Sets were employed. Because of the fact that no duplicate entries should be present in the application because all of the objects have unique identifier the `java.util.HashSet` was mostly used.

5.4.1.1 User

This class is common to all users of the system and contains general attributes describing a user of I-Test. User is a super class of TestManager, Administrator and Student classes. This ISA hierarchy was mapped to SQL as four separate tables because of the fact that most of the relations are unique to the subclasses. The mapping used is cleaner but brings also minor performance drawback. However, this can be simply changed in case of any problems.

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>user_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>firstName</td>
<td>String</td>
<td>first_name</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>lastName</td>
<td>String</td>
<td>last_name</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>userName</td>
<td>String</td>
<td>user_name</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>password</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>email</td>
<td>String</td>
<td>email</td>
<td>VARCHAR (70)</td>
</tr>
<tr>
<td>role</td>
<td>String</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lastLogin</td>
<td>Access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>access</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1 Properties of class User

5.4.1.2 Student

Student class inherits most of the properties from generic User. It furthermore contains the date of birth of the student. Each student in the system is owned by single test manager.
<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>student_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>birthDate</td>
<td>Date</td>
<td>birth_date</td>
<td>DATE</td>
</tr>
<tr>
<td>studentGroups</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>manager</td>
<td>TestManager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testInstances</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2 Properties of class Student

5.4.1.3 TestManager

TestManager class does not have any real properties and only contains containers to store other objects which test manager owns (students, student groups, tests, question banks).

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>manager_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>tests</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>students</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>studentGroups</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>questionBanks</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3 Properties of class TestManager

5.4.1.4 Administrator

The Administrator class only inherits all properties from User and does not have any own member variables.

5.4.1.5 StudentGroup

Class StudentGroup is used as a container to group students to facilitate the delivery of assessments. StudentGroup only has a name and a set of students which belong to it. Unlike the case of question banks, there is no default student group created because students need not to be members of any group in order to exist in the system.

5.4.1.6 TestInstance

This class represents a Test which was delivered to a student. Every instance of a test contains set of randomly picked questions that were selected from question banks which are contained in a test.
### Table 5.4 Properties of class TestInstance

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>test_instance_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>startTime</td>
<td>Date</td>
<td>start_time</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>endTime</td>
<td>Date</td>
<td>end_time</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>score</td>
<td>Integer</td>
<td>score</td>
<td>INT (4)</td>
</tr>
<tr>
<td>maxScore</td>
<td>Integer</td>
<td>max_score</td>
<td>INT (4)</td>
</tr>
<tr>
<td>test</td>
<td>Test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>student</td>
<td>Student</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testInstanceQuestions</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.4.1.7 Test

The class Test represents an assessment. It contains many properties but some of them are not used in current version (startTime, allowedHosts and displayAllQuestions).

The use of start time was removed from the implementation. This reduces the system load when all of the students try to begin an assessment at the same time. Instead the publishTime was used to allow the author to hide an assessment and to emulate startTime in case that it is needed.

Current version also does not allow limiting the access to the system to certain hosts or IP addresses only. Also it is only possible to display all questions on single page.

### Table 5.5 Properties of class Test

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>test_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>name</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>startTime</td>
<td>Date</td>
<td>start_time</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>duration</td>
<td>Integer</td>
<td>duration</td>
<td>INT (4)</td>
</tr>
<tr>
<td>publishTime</td>
<td>Date</td>
<td>publish_results</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>endTime</td>
<td>Date</td>
<td>end_time</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>displayResults</td>
<td>Boolean</td>
<td>display_results</td>
<td>BOOL</td>
</tr>
<tr>
<td>allowedHosts</td>
<td>String</td>
<td>allowed_hosts</td>
<td>VARCHAR (255)</td>
</tr>
<tr>
<td>randomised</td>
<td>Boolean</td>
<td>randomised</td>
<td>BOOL</td>
</tr>
<tr>
<td>displayAllQuestions</td>
<td>Boolean</td>
<td>display_all_questions</td>
<td>BOOL</td>
</tr>
<tr>
<td>instructions</td>
<td>String</td>
<td>instructions</td>
<td>TEXT</td>
</tr>
<tr>
<td>manager</td>
<td>Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testQuestionBanks</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>testInstances</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.4.1.8 Question

This class represents an assessment item. Question contains only the basic properties and all new question types which will be implemented in the future would inherit from this basic
class. In this model the question contains *timeLimit* property which is not used because the delivery of assessment on question per page basis is not supported.

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>question_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>title</td>
<td>VARCHAR (255)</td>
</tr>
<tr>
<td>points</td>
<td>Integer</td>
<td>points</td>
<td>INT (4)</td>
</tr>
<tr>
<td>timeLimit</td>
<td>Integer</td>
<td>time_limit</td>
<td>INT (4)</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>description</td>
<td>TEXT</td>
</tr>
<tr>
<td>type</td>
<td>Integer</td>
<td>type</td>
<td>INT (4)</td>
</tr>
<tr>
<td>questionBank</td>
<td>QuestionBank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6 Properties of class Question

### 5.4.1.9 QuestionBank

QuestionBank class represents a container for Questions. Every newly created TestManager has a default bank that cannot be removed and this bank is marked as default bank by setting the *defaultBank* property to true.

Question banks can be in two states - unlocked or locked. A Question bank becomes locked when it is included into an assessment. Locked question banks cannot be removed. Questions contained within them cannot be deleted and their modification is very limited.

<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>question_bank_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>name</td>
<td>VARCHAR (50)</td>
</tr>
<tr>
<td>defaultBank</td>
<td>Boolean</td>
<td>default_bank</td>
<td>BOOL</td>
</tr>
<tr>
<td>locked</td>
<td>Boolean</td>
<td>locked</td>
<td>BOOL</td>
</tr>
<tr>
<td>manager</td>
<td>TestManager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>questions</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.7 Properties of class QuestionBank

### 5.4.1.10 TestQuestionBank

TestQuestionBank class represents the relation between an assessment and question banks included into this assessment. Its only real property is number of questions used from the question bank. The rest are just relations to corresponding test and question bank.

### 5.4.1.11 TestInstanceQuestion

This class is used to represent the relationship between an instance of a test which was assigned to certain student and questions included within the test. The *position* property is used to determine the order of questions in a test. During the scoring of a test the
answeredCorrectly property is being set to true. This property is then used when displaying results of a test so that it is not necessary to evaluate student’s responses again.

<table>
<thead>
<tr>
<th>Property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Long</td>
<td>test_instance_question_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>Position</td>
<td>Integer</td>
<td>position</td>
<td>INT (4)</td>
</tr>
<tr>
<td>answeredCorrectly</td>
<td>Boolean</td>
<td>answered_correctly</td>
<td>BOOL</td>
</tr>
<tr>
<td>testInstance</td>
<td>TestInstance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responses</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>Question</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.8 Properties of class TestInstanceQuestion

5.4.1.12 Answer

Answer represents one of the choices in multiple-choice assessment items. The answer in this model contains the points property which is not used in the application and the question score is determined only by the points property of a question.

<table>
<thead>
<tr>
<th>Property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Long</td>
<td>answer_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>Points</td>
<td>Integer</td>
<td>points</td>
<td>INT (4)</td>
</tr>
<tr>
<td>Feedback</td>
<td>String</td>
<td>feedback</td>
<td>TEXT</td>
</tr>
<tr>
<td>Correct</td>
<td>Boolean</td>
<td>correct</td>
<td>BOOL</td>
</tr>
<tr>
<td>answerText</td>
<td>String</td>
<td>answer_text</td>
<td>TEXT</td>
</tr>
<tr>
<td>Question</td>
<td>Question</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responses</td>
<td>Set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.9 Properties of class Answer

5.4.1.13 Response

This class is used to store student’s response for a particular question. Because only multiple choice assessment items are supported, only the Boolean value indicating whether the user has marked the choice or not is stored in the database. It is assumed that the new types of questions will also inherit new response types from this parent class.

5.4.1.14 Access

This class is used to track and monitor users’ access to the system and would find further use when the option to limit access to the whole system or assessments from certain hosts or networks only.
<table>
<thead>
<tr>
<th>property</th>
<th>Java type</th>
<th>SQL field</th>
<th>SQL type</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Long</td>
<td>access_id</td>
<td>INT (8)</td>
</tr>
<tr>
<td>ipAddress</td>
<td>String</td>
<td>ip_address</td>
<td>VARCHAR (30)</td>
</tr>
<tr>
<td>time</td>
<td>Date</td>
<td>access_time</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>testInstanceQuestion</td>
<td>TestInstanceQuestion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>answer</td>
<td>Answer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.10 Properties of class Access
5.5 Data Model

Data model of the application was created from the object relational mapping by Hibernate. The presented model only describes the relations between the entities and also foreign key constraints. The model is only included to demonstrate the way how ORM framework maps classes to SQL tables.

![E-R Diagram of I-Test](image)

Fig. 5.7 E-R diagram of I-Test

Notation:
- **PK** primary key
- **FK** foreign key
- ← relation between two tables
6 Testing of the Application

Testing and analysis are the most important parts of software development process. I-Test is in the phase of working prototype and therefore significant efforts have been made in testing of its functionality. White-box tests were performed during the whole development of the application. The application can be divided into three parts according to user roles of the system. These parts were tested separately in the first phases and subsequently, the system was tested as a whole during integration tests.

6.1 General Testing

Due to the possible problems with compatibility of web browsers the application was fully tested under Windows XP with Microsoft Internet Explorer 6.0 and Mozilla Firefox 1.0.7. Other tests were performed under Windows 98 with Internet Explorer 5.0. All the pages were displayed correctly and the application worked without any problems. These browsers according to my personal observations and other statistics [34], more than 90% people use these browsers. It is believed that owing to the simple design of I-Test, there should not be any major problems with other browsers used nowadays such as Opera, Netscape, etc.

Finally further tests were performed under operating system Linux (Knoppix 4.0.2 distribution) with Konqueror 3.4.1, Mozilla Firefox 1.0.6 and text browser Lynx. Although the layout of the pages was rendered correctly, in Konqueror and Lynx, some of the Czech characters were displayed with missing diacritic. It is believed that this issue was caused due to missing or incorrectly installed fonts in Knoppix and that it is not related to the application itself. Quite surprisingly the application was easily navigable even without the use of graphical web browser in text only browser Lynx. Testing the application in browsers without any graphic elements and without colour information is very important for compatibility with text readers and special browsers for blind people because it displays the pages in a similar way as these programs do.

To verify the influence of database system used, the functionality of I-Test with MySQL 4.0 database had to be tested. Thanks to the use of Hibernate, the migration to another database was very simple and required only modification of several configuration files and installation of appropriate MySQL JDBC drivers. Hibernate provides the capability of creating the structure of the database according to the selected dialect of the database and therefore the only additional step was creation of administrator account. At first sight the application
responded much faster than with use of PostgreSQL but from the reasons mentioned above PostgreSQL is recommended to be used.

The first black-box tests were performed on local network with two client computers. Although I-Test was installed on a laptop (detailed configuration can be found in Table 6.1) instead on a powerful web server, the application performed well without any glitches. Testing discovered mostly minor issues in user interface which were subsequently fixed.

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Centrino 1.4 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>1.2GB</td>
</tr>
<tr>
<td>HDD</td>
<td>40GB</td>
</tr>
<tr>
<td>OS</td>
<td>Windows XP Home SP2</td>
</tr>
</tbody>
</table>

Table 6.1 Configuration of laptop used for testing

6.2 Beta Testing with Assistance of Students

To test the performance of the application and its features, an assessment was taken on small group of students. The headmaster of Skolni primary school offered the possibility of testing the system on their computer cluster. With help of Kvetoslava Bicikova, a teacher from this school, an assessment of Czech language and literature for third grade children was created, consisting of seven parts (each part contained several questions). The questions were randomly picked from question banks to form the final test comprising of 20 questions. This assessment was delivered to 13 students whose accounts were imported into I-Test prior to the assessment. The assessment provided no time limit so that no stress was placed on the children as this was their first time doing an online assessment. However, the test was set to the end by the end of the lesson. Results of the assessment were not published right after completion of the test but were released after all the children submitted the assessment by modifying the properties of the assessment.

The Application was once again installed only on a laptop used in previous tests which was connected to local network. Client computers were not of the same type using following operating systems: Windows XP, Windows 2000 and Windows 98. Web browsers used were Internet Explorer 6.0 and 5.5. All the clients were accessing the server using secured connections by HTTPS.

The system performed well without any errors during the whole session which lasted more than 45 minutes. Although the application was set to debug mode and was thus logging all performed database queries, the responses were very fast without any delays.
The use of different workstations with different display devices affected the objectivity of the test. Some of the display devices were old CRT monitors but the others were modern LCDs. The use of old CRT monitors led to incorrect answers where the letter ć was misinterpreted as the letter r. Therefore the same workstations must be used not to handicap certain users and maintain the objectivity of the test.

Although the children were of ages 8 and 9, nobody had any difficulties and after a brief introduction, they were able to follow the instructions, login to the system, complete the test and display correct results. Randomisation of questions and answers appeared as a good choice because the proximity of workstations significantly decreased the possibility of cheating. All the children were asked to verify that they answered all the questions before submitting the assessment but most of them skipped some of the questions and thus lost some points. This fact led to minor modifications in the implementation and a warning message notifying users to make sure that they answered all the questions is displayed prior to the submission of the assessment to prevent this issue.

In general the children liked this form of assessment which seemed to them more like a game than a formal exam and thus probably did not cause too much stress which is inherent in the latter. The assessment was repeated two times to observe the improvement of the children. The second attempt was completed by almost all of the children except two. Thanks to the random generation of questions from question banks none of the respondents was delivered an assessment consisting of exactly the same questions but still most of the questions overlapped because of insufficient number of different versions stored in the banks. Not surprisingly all of the children who completed the test achieved the same or better result. The average improvement was about 13%.

The examination of test statistics revealed that the test was designed quite well.

<table>
<thead>
<tr>
<th>Total number of delivered tests</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of submitted tests</td>
<td>13</td>
</tr>
<tr>
<td>Number of questions in test</td>
<td>20</td>
</tr>
<tr>
<td>Maximum number of points</td>
<td>20</td>
</tr>
<tr>
<td>Average test score</td>
<td>12.46</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.39</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.28</td>
</tr>
<tr>
<td>Difficulty</td>
<td>62.31%</td>
</tr>
</tbody>
</table>

*Table 6.2 Sample assessment statistics*
To conclude the results of testing, the system has proved its functionality but requires further verification with more concurrently working students and test managers to discover possible existing bottle necks. The experience with online assessment performed with small children revealed that it can be successfully used to promote learning and revise the curriculum although its use for real assessment is considerable.

6.3 Known Issues
There is one known problem in the application which occurs sometimes when a user does not wait until his requested is processed. That is the page containing its result gets fully loaded and tries to display another page. This can result in displaying of application error. To avoid this problem it is recommended to wait for each request to be completed. This issue was not solved due to time reasons and because the fact that it does not affect normal use of the system is not considered as crucial.
7 Future Work

Due to time limitations and complexity of the task I-Test is still in the state of working prototype and therefore many features must have been implemented before this system can be released. The aim of this chapter is to describe future steps which have to be undertaken in order to finalise this product.

One of the first steps which should be done is to increase the number of supported assessment items because current version of I-Test supports only multiple choice questions. Other question types should be implemented according to the results of analysis in chapter 3.8.3.

I-Test can be used as web based assessment service so it might be desired to expand its features and to allow hosting of assessments. This step would require implementation of usage statistics to track the use of the system by test managers and constraints used to limit the numbers of assessments delivered by a test manager. In addition, some accounting system enabling test managers to buy additional credit to increase these limits would be much desired. Similar approaches as the one used by Hosted Test (3.7) could be introduced. Implementing these features would create new user roles in the system like accountants who would be responsible for dealing with payments.

The lack of support for images and other multimedia files in assessments and questions is not coincidental. Although this feature was implemented in all other analysed systems, its implementation is complex. This functionality had been considered for a long time but was not implemented. The best solution suggested to allow the use of multimedia objects within I-Test is the creation of some secure virtual file system which would also allow sharing of these resources.

An emphasis was placed on the compliance with main e-learning standards in this text but none of those standards is supported by I-Test. Implementing the support for exporting and importing learning assessments and assessment items packaged according to the IMS packaging specification would be one of the next goals.

The testing of the system in real environment revealed some fresh ideas which were not considered during the design of the application. To further increase the objectivity of assessment, scoring assessments of students suffering from various disabilities like dyslexia, dysgraphia, etc. should be a priority. The current release does not consider this fact.
Future releases of the application could also contain some other improvements like presenting the user with the possibility of saving the assessment during the work to reduce the risk of losing it due to some unexpected system or power failure. The saving could be implemented manually or automatically.

Also the support of email communication from within the system is desirable to allow test managers to easily send messages to students or groups of students and notify them about upcoming events.

User interface of the application is suitable for use with small numbers of students, questions and tests. Some improvements like enabling sorting, filtering and searching of these entities should be implemented to make their management more comfortable.
8 Conclusion

The goals of this master thesis were to acquaint myself with the problems of online assessment and suitable technologies, and to design, implement and test system allowing preparation of online assessments. I familiarised myself with the basics of the theory of assessment and some issues which are related to the preparation of multiple choice questions. The foundations I acquired in this phase helped me during the analysis and evaluation of some of the applications dealing with assessment of students.

There are many assessment systems on the market and it is not possible to analyse all of these systems. It was necessary to choose representative sample for more detailed evaluation. At the commencement of my work, the Czech Technical University in Prague was performing trial runs of its new e-learning platform Microsoft Class Server. Consequently, I decided to focus on assessment systems offered by some of the most popular LMSs. I believed that this decision did not reduce the researched scope and that assessment modules comprising all major LMSs form good representative sample. To ensure the objectivity of the analysis, other systems belonging to different categories were included in the survey. The detailed analysis of existing systems discovered all the important features of a good assessment tool.

During the design time several technologies facilitating the development and reusability of the final product were considered and the most suitable ones were used to implement simple assessment application allowing creation and delivery of online assessments. Despite having more than four years of practice with PHP and MySQL technologies together with the fact that most of the evaluated systems use this, I decided to implement the application in Java and PostgreSQL. This decision made the development of the application more difficult in the beginning because it was necessary to get accustomed to the new technologies. However, at the end the benefits of object oriented approach, the use of design patterns and frameworks outweighed the initial difficulties.

Testing of the application was performed during the whole development and resulted in beta testing of the software with assistance of children from primary school. The current state of the application allows its release to the public however at least some of the improvements mentioned in chapter 7 should be done.

Working on this thesis allowed me to focus deeply on the problem of online assessment in e-learning and to acquire new programming techniques and technologies which I found
valuable and useful. These led to improvement of my analytical and programming skills. The final product is a working application which was tested and can be used in e-learning to assess learners. Thanks to its simple usage not requiring any further training and only little practice it can find use in environments with special needs like assessment of smaller children or seniors.
9 Bibliography


[27] Apache OJB: http://db.apache.org/ojb/
[34] Web browser statistics: http://www.w3schools.com/browsers/browsers_stats.asp
A Installation Manual

This manual describes installation of I-Test and all required software packages on a single computer. Due to the fact that I-Test was developed in Java it is platform independent. It can be deployed on most common operating systems like Microsoft Windows, Linux or Unix without any difficulties. However, only the installation under Microsoft Windows will be described in this manual.

Because of the fact that I-Test is a prototype only the basic installation of the system showing its capabilities and functionality will be described here. Linux or Unix are more suitable operating systems for running web servers. Installation of database server and web server on separate machines would increase the overall performance of the application.

All required software packages can be found on the attached CD. Latest versions of these packages can be downloaded from the Internet but it is not guaranteed that I-Test will work with them. The recommended versions of all packages and libraries are listed in prerequisites.

Prerequisites:

- Java Runtime Environment 1.5.0
- Apache Tomcat 5.5.9
- PostgreSQL 8.0.3

A.1 Java Runtime Environment

At first it is necessary to make sure that Java Runtime packages are installed in the system. This can be done by launching Windows Control Panel. The presence of Java icon on the panel indicates that some version of Java Runtime is installed. Double click the icon and then in the General tab click the About... button to determine the version of the runtime.

If the installed version is 1.5.0 you can continue with installation of other components otherwise Java Runtime must be installed. The installation file can be found on attached CD in following directory: Support\JRE 1.5\jre-1_5_0_04-windows-ia586-p.exe or the latest version can be downloaded from the Internet (http://java.sun.com/j2se/1.5.0/download.jsp).
A.2 Apache Tomcat

The installation of Apache Tomcat is straightforward and will not be described in this text in detail. To begin the installation, launch the installation file and follow the instructions. Installation file of Apache Tomcat web server can be found on attached CD in following folder: \Support\Tomcat 5.5.9\jakarta-tomcat-5.5.9.exe

To verify the installation point your favourite web browser to: http://localhost:8080/

A.2.1 Configuring Tomcat

For testing purposes of I-Test Tomcat does not require any further configuration however it is recommended to enable and use SSL to increase the security of I-Test. Detail instructions on enabling SSL on Tomcat can be found in [35].

A.2.1.1 Installation of JDBC Driver

It is necessary to install JDBC driver in order to connect to PostgreSQL database (in case that you want to try to use different database appropriate JDBC driver must be copied into appropriate folder). The driver can be found on the attached CD in following folder: \Libraries\postgresql-8.0-312.jdbc3.jar. Copy this file into $TOMCAT_HOME\common\lib$. Where $TOMCAT_HOME$ refers to the folder where Tomcat was installed (typically: C:\Program Files\Apache Software Foundation\Tomcat 5.5)

A.3 PostgreSQL

Prior to the installation of database engine make sure that your system has at least one partition formatted with NTFS file system. PostgreSQL for Windows requires partition of this type to be present in the system.

The installation files of PostgreSQL database are on the attached CD in following folder: \Support\PostgreSQL\postgresql-8.0.3.zip

To install the database unzip the content of postgresql-8.0.3.zip into some temporary folder. This can be done using integrated Windows XP feature or using WinZip [32]. When the files are unzipped, launch the postgresql-8.0.msi file. Proceed through the installation wizard with default settings.
A.3.1 Configuring PostgreSQL

Complete configuration of PostgreSQL is beyond the scope of this text hence only the steps required to make the I-Test system running will be described. I-Test requires single user account and database to be created. You can skip following steps if you know how to do this.

To manage PostgreSQL database launch *pgAdmin III* tool that can be found under the *Start* menu. When the program is loaded double click the item named: “PostgreSQL Database Server 8.0...” on the left pane containing list of all available servers. Enter the password of user “postgres” you set during the installation. A view of following objects contained within the server should be displayed: Databases, Table spaces, Groups and Users.

A.3.1.1 Creating User Account

Click on the *Users* item and select *New User* command. Enter Username and password and click OK to confirm addition of new database user. It is recommended to name the user *itest*.

![Fig. A.1 Creation of new user account in pgAdmin](image)

A.3.1.2 Creating Database

To create new database click on the Databases item with right mouse button and select the *New Database* command.
Fig. A.2 Creation of new database in pgAdmin

- Fill in the name of the database. It is recommended to use the name *itest*.
- Make sure that database encoding is set to UNICODE.
- Set the owner of the database to the username which was used in previous step.

Fig. A.3 Setting properties of new database
A.3.1.3 Creating Initial Database Structure

It is necessary to create tables used by I-Test and create the account of administrator of the system with default password. This can be done by executing SQL script which can be found on the attached CD in following file: SQL\create.sql. To execute this script launch command prompt, type following command at the prompt and press Enter:

```
"POSTGRES_HOME\bin\psql.exe" -f CD_DRIVE\SQL\create.sql -U DB_USER DB_NAME
```

Where:

- POSTGRES_HOME is path to PostgreSQL installation. By default this should be: “C:\Program Files\PostgreSQL\8.0”
- CD_DRIVE is the letter of your CD/DVD-ROM drive (typically: D: or E: )
- DB_USER the name of the account which was created in A.3.1.1 (itest by default)
- DB_NAME the name of the account which was created in A.3.1.2 (itest by default)

Example of the command:

```
C:\Program Files\PostgreSQL\8.0\bin\psql.exe -f D:\SQL\create.sql -U itest itest
```

After executing this command a prompt asking for a password of user DB_USER appears. Enter the correct password for given user to initiate creation of the structure of the database.

A.4 Deploying I-Test

Deployment of I-Test is very simple and does not require many steps. The application is packed in WAR file. Copy following file: WAR\itest.war into webapps directory located under TOMCAT_HOME. Tomcat should then load the application automatically. In case that further help is required, please refer to Tomcat documentation. [31]

Following steps described in chapter A.4.1 can be skipped if the DB_USER and DB_NAME variables were kept to their default values (itest, itest).

A.4.1 Configuring Database Connection

In case that default names were not used the configuration of database connection must be altered. Edit the configuration file of persistency layer hibernate.cfg.xml which is located
under: `TOMCAT_HOME\webapps\itest\WEB-INF\classes` and modify following lines accordingly:

```xml
<property
  name="hibernate.connection.url">jdbc:postgresql://localhost/DB_NAME</property>
<property name="hibernate.connection.username">DB_USER</property>
<property name="hibernate.connection.password">DB_USER_PASSWORD</property>
```

**A.4.2 Configuring I-Test to Require SSL**

The web application can be configured to require SSL connections only. This can be set in the `web.xml` file.

Uncomment following lines of code located at the end of the file:

```xml
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Secured pages</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

**A.5 Logging into I-Test**

After successful completion of previous steps I-Test is now ready to be used. Point your favourite browser to: [http://localhost:8080/itest](http://localhost:8080/itest). Login screen should appear after some time.

It may take up to a few minutes for the first time for I-Test to initialize. The login details of administrator are following:

- User name: admin
- Password: admin

It is recommended to change the password immediately after the first login!
B User Manual

I-Test is web based assessment system allowing creation and delivery of online tests. The system allows teachers to easily import data about their students into the system, create multiple choice questions, organise these questions into question banks and compose assessments which can be delivered to students or defined student groups. The system also generates test reports and detailed statistics which can be used to monitor the quality of the assessment.

Security is considered as very important in the system, please note that all inappropriate actions which could lead to unauthorised access to confidential information are logged and can be traced by administrators of the system.

The system is using JavaScript to display some warnings and notices and in case of a timed test to display a timer. User is not required to have JavaScript enabled in his browser in order to be able to work with the application but this step is recommended to make the work with the application more comfortable.

B.1 Common Features

This section describes general features which are common to all users of I-Test and therefore should be read by everyone.

The layout of the application is very simple and consists of three parts: header, menu, content and footer.

- **Header** contains only the logo of the application in this version
- **Menu** is used to navigate through the system and is the key component
- **Content** displays the content of pages accessed using the main menu and presents users with further available commands
- **Footer** is not used in current release

9.1.1 Logging into the System

Before starting to work with the system it is necessary to log in. This can be done by entering your user name and password in appropriate text boxes on the login screen displayed on Fig. B.1.
After successful login to the system, user is presented with welcome screen displaying his full name, user role, date, time and place of last login. In case that any of these details are incorrect, please contact the person with higher privileges who created your user account and is responsible for its maintenance. Pay attention especially to the last login date and time to make sure that nobody has abused your user account during the period you have not used the system.

9.1.2 Logging out of the System
It is recommended to logout of the system after finishing your work to reduce the risk of unauthorised use of your account. The session remains active until the logout button is pressed, the web browser window is closed or after six hours of inactivity. This long inactivity period was set to allow performing of long lasting examinations. After logging out, the system should redirect user back to login screen. It is recommended to close your web browser after logout to ensure that no sensitive information can be accessed by other users.

9.1.3 Using Back Button of Web Browser
Use of the back button in browser to navigate the site is not recommended because in some cases it may not work as expected. In case that user presses the back button after submitting a form the browser has to resend the data which has been already submitted in order to get
backwards. This can in some cases result in an error especially when adding new objects to the system. Therefore it is recommended to use this button to get back. Incorrect use of the back button cannot cause any system failure.

9.1.4 Locked Items

Sometimes the user can encounter following symbol indicating that the object is locked or that the action is not available at the moment. In case of locked objects the symbol is followed by brief description giving the reason why the object is locked and what actions should be performed to unlock it.

9.1.5 Filling in Forms

When creating or editing any forms. It is necessary to fill in all fields marked with * otherwise validation errors occur and user will be asked to correct these errors. Some fields require values of certain types to be filled in. In case that incorrect input is entered validation errors appear again until correct values are provided.

User name in I-Test must contain at least 5 characters and must be unique in the system (user name is only entered by Administrators and Test Managers when creating new user account and cannot be modified later). In case that the same user name already exists, validation error appears notifying about this problem. To fix this it is necessary to choose a different user name.

Password can contain any characters and must be at least 5 characters long. Maximum lengths of the fields are limited by web browser. For fields requiring specially formatted input this is usually mentioned in a comment placed after the text box.

Successful completion of desired operation is indicated by confirmation message displayed at the top of the content frame usually saying “Changes were saved successfully” or “User was created successfully”, etc.

9.1.6 Changing User Details and Settings

User details and settings can be accessed and modified by pressing the Settings button in main application menu. Please note that it is not possible to modify all of the user details. Users are normally allowed to change their e-mail address and password.
B.2 Administrator’s Tasks

The user interface of I-Test offers only limited functionality for administrators but it is assumed that they have full access to the web server and database and can use this to administer the system. By default there is only single administrator account in the system but more accounts can be created in large deployments.

An administrator can choose following actions from the main application menu.

- **Home** – displays initial welcome screen
- **Test Managers** – displays module allowing to create, edit and delete Test Managers
- **Access log** – displays list of all accesses to the system
- **Settings** – allows to edit user details and settings
- **Logout** – logs user out of the system
- **Help** – displays online help in new browser window

**B.2.1 Test Managers Administration**

This section of the system is responsible for creation, modification and deletion of user accounts of Test Managers. Sample screen is displayed on Fig. B.3. To create a new account
of Test Manager press the *Add Test Manager* button located at the top of the main body. To delete or edit an account find it in the list of Test Managers and press appropriate buttons.

*Please note that deletion of Test Manager causes cascade delete and removes all objects owned by the manager including students, questions, assessments, etc.*

If JavaScript is enabled in your browser a warning dialogue is displayed to confirm deletion of the user otherwise it is removed immediately.

---

**Fig. B.3 List of Test Managers**

**B.2.1.1 Creation and Editing of an User Account**

A new account can be created by pressing the *Add test manager* button. A dialogue similar to the one on Fig. B.4 appears when adding or editing a user. All validation errors must be fixed before the modification can be stored into the database.
B.2.2 Working with Access Log

To display the access log press the Access log button in the main menu. Access log allows system administrators to monitor the usage of the system and to track some possible problems related to unauthorised accesses. However the log files generated by the application provide more details related to security threats. The access log can grow very quickly and therefore it is recommended to delete old records from time to time. This can be done by pressing the Delete old records button. This causes that only the latest access for every user remains and all the other ones are removed permanently.
Test Manager’s Tasks

Test manager is the key user role in the system and is responsible for the management of students, student groups, questions, question banks and assessments. A test manager can choose following actions from the main application menu.

- **Home** – displays initial welcome screen
- **Students** – allows to view, add, edit, import and delete students
- **Groups** – allows to view, create, edit and delete student groups
- **Questions** – allows to view, create, edit, duplicate and delete questions
- **Banks** – allows to view, edit, create, edit, duplicate and delete question banks
- **Tests** – allows to view, create, edit, delete, and deliver assessments and also to display assessment results and statistics
- **Logout** – logs user out of the system
- **Help** – displays online help in new browser window

### B.3.1 Managing Students

Test manager can work with students in a similar way as Administrators can work with Test Managers. Please refer to section B.2.1 for further details. In case of students it is necessary to
fill in one additional parameter the date of birth of the student. This date must be entered in following format: dd/MM/yyyy.

Where:

- **dd** is 2 digit representation of a day in month
- **MM** is 2 digit representation of the month
- **yyyy** is 4 digit representation of the year of birth

### B.3.1.1 Importing Students

Test Manager has the possibility to import large numbers of students from text file instead of adding the students manually. This feature is accessible by pressing the *Students* button from main menu and then clicking on *Import students* button. Similar dialogue as the one displayed on Fig. B.6 should appear allowing user to select input file and choose student groups into which the newly imported students are added.

![Fig. B.6 Importing students](image)

In case that some errors occur during the import no records are added before all the errors are fixed.
Format of the Imported File

The input file must have exactly the same format as the first header line which is omitted during the import and be in the UTF-8 character encoding:

username,first name,last name,email,birth date,password

Each record is then put on a separate line. A sample input file follows:

username,first name,last name,email,birth date,password
jbicik,Josef,Bičík,josef.bicik@atlas.cz,11/08/1980,password1
fnovak,František,Novák,novak@email.cz,23/06/1981,password2

B.3.1.2 Delete all Students

To delete all students owned by the currently logged in Test Manager go to the Students section and press the *Delete all students* button. Although a confirmation dialogue is displayed, it is recommended to use this action with caution because deletion of all students also discards all their assessments and scores.

B.3.2 Working with Questions and Question Banks

Questions in I-Test can be only of multiple-choice type in the current version and are organised into Question Banks. Every Test Manager has a default question bank but it is recommended to create more question banks. To create a new question, click the *Add new question* button in the *Questions* section. Question has the following attributes:

- **Question type** – can be either multiple responses or single response only question
- **Question** – text of the question
- **Points** – number of points representing the weight of the question
- **Detail** – detailed description of the question
- **Question bank name** – the name of question bank into which the question is stored
- **Answers** – one to eight choices with corresponding feedbacks. The feedback is displayed in case that the choice is checked by a student. In case of questions with single correct response only exactly one of the answers must be marked as correct.

Questions that have already been included in assessments cannot be fully modified and certain properties are locked. The only solution to unlock them is to make sure that the question bank which contains the question is not included in any assessment.
B.3.3 Creating and Editing Assessments

To create or edit an assessment navigate to the Tests section accessible from the main menu. In this section click either on Add new test button or appropriate Edit button. A dialogue similar to the one displayed on Fig. B.8 appears. Description of available assessment properties follows:

- **Name** – the name of the assessment which is displayed to users
- **End date & time** – all assessments must be submitted before this date and time
- **Duration** – duration of an assessment in minutes
- **Publish date & time** – date and time when the assessment is visible to students
- **Display test results** – indicates if assessment results are published to students when they finish the test (can be modified later after all students submit the assessment)
- **Instructions** – Instructions displayed at the top of the test
- **Question banks** – list of question banks that can be included into the assessment. Exact number of questions which are randomly chosen from each question bank can be specified for each question bank.

End and publish dates must be entered in following format: dd/MM/YYYY HH:mm:ss
Where:

- $dd$ is 2 digit representation of a day in month
- $MM$ is 2 digit representation of the month
- $yyyy$ is 4 digit representation of the year of birth
- $HH$ is 2 digit representation of hour in 24 hour format
- $mm$ is 2 digit representation of minutes
- $ss$ is 2 digit representation of seconds

Please note that all question banks and all questions which are included in an assessment become locked and cannot be fully edited or deleted. However, it is possible to duplicate them to create a copy which can be altered.

**B.3.4 Delivering Assessments**

To deliver an assessment click the *Deliver* button corresponding to appropriate assessment in the list of tests which can be accessed by clicking the *Tests* button from the main menu. User can then select groups or single users for delivery of the test. Multiple selections can be made by holding the *Shift* or *Ctrl* keys when selecting items. To make the test undelivered ensure that only [none] is selected in both *Groups* and *Students* list boxes.
Fig. B.9 Delivering an assessment

Please note that removing student from the delivery group also discards the test results gained by this user. Therefore it is necessary to change the selections carefully.

Tests which have been already delivered to students cannot be fully edited and only certain properties like displaying of the results, test instructions and the name of the assessment can be modified.

**B.3.5 Displaying Results and Statistics of an Assessment**

Results of an assessment can be accessed from the *Tests* section by clicking on the *Results* button corresponding to given assessment. The results are presented using a table which can be copied to other application using clipboard. Tests of individual students can be displayed from the screen with displayed results by clicking on the *View test* button. This opens dialogue similar to the one displayed on Fig. B.14.

To display the statistics of an assessment click on the *View test statistics* button. Statistics are based on submitted tests only and present values of some of the factors affecting validity of an assessment. Statistics related to specific questions are displayed in a table below the overall assessment statistics. Sample assessment statistics are displayed on Fig. B.10.
B.4 Student’s Tasks

A student can choose following actions from the main application menu.

- **Home** – displays initial welcome screen
- **Tests** – displays all available tests and their scores
- **Settings** – allows to edit user details and settings
- **Logout** – logs user out of the system
- **Help** – displays online help in new browser window

Student can display all available tests and also test scores by pressing the **Tests** button from the main menu. Sample list of assessments is displayed on Fig. B.11.
Fig. B.11 List of available assessments

B.4.1 Taking an Assessment

Open tests can be taken by pressing the Take test button. Please note that after pressing this button the test is launched and when its duration was set the timer is started and cannot be paused. Tests with end date and time specified do not contain a timer which can be found on Fig. B.13 but must be submitted before the end time of the tests.

Tests with specified duration have automatic timer (JavaScript must be enabled). This timer counts down the available time to complete the assessment. If the assessment is not submitted manually by pressing the SUBMIT TEST button this is done automatically when the timer reaches zero.

Before submission of the test a warning message is displayed asking user to verify that all questions have been answered (this is not the case of automatic submission).
Important note: Late submissions are not taken into account and such tests are scored by zero points. Before starting an assessment it is recommended to verify that your system time corresponds to server time which is only considered when calculating the submission time.

**B.4.2 Displaying Results**

After completion of an assessment the correct answers can be displayed by pressing the *Results* button. This button is only available if this option was enabled by Test Manager.

Correctly answered questions have light green background and incorrectly answered questions have pink background and their score is set to 0. Correct choices which should have been checked are marked by ✔️. Optionally a feedback can be displayed bellow the question. Sample assessment results are displayed on Fig. B.14.
Fig. B.14 Completed assessment with correct answers
C Content of Attached CD-ROM

A brief description of the content of the attached CD and its structure follows:

/Doc – this master thesis available in PDF and Microsoft Word formats

/JDeveloper 10 – IDE from Oracle which was used to develop I-Test

/Libraries – contains all libraries used by I-Test
   /hibernate-3.1rc2.zip – Hibernate persistency framework
   /mysql-connector-java-3.1.6-bin.jar – MySQL JDBC driver
   /opencsv-1.0-src-with-libs.tar.tar – library for parsing CSV files
   /postgresql-8.0-312.jdbc3.jar – PostgreSQL JDBC driver
   /struts-1.2.7.zip – Struts framework

/Sources – source codes and project file

/SQL – SQL scripts to create initial database structure
   /create.sql – SQL script to create tables used by I-Test for PostgreSQL database

/Support – contains software packages which have been used during the development process
   /JRC-Editor – Application used to edit Resource bundles
   /JRE 1.5 – Java Runtime Environment
   /PostgreSQL – database server used by I-Test to store data
   /Tomcat 5.5.9 – web server used to host I-Test application

/WAR – package containing the whole application