Summer 2007

Web Based Candidate Assessment System

Noel Kyne
Regis University

Web Based Candidate Assessment System

Noel Kyne 10th June 2007
Noel.Kyne@gmail.com

Regis University

School for Professional Studies

Master of Science in Software and Information Systems

Abstract

Devplex Technologies Limited is a privately owned company based in Galway Ireland. They have been operating for over two years and currently undertake contract projects for the travel and financial industries. The projects are varied and a wide range of skills are necessary. Devplex Technologies are currently undergoing expansion and intend to hire a number of new employees with varying levels of experience. Devplex Technologies also employ a high number of contractors, with varying skills and contract periods range from one month to twenty four months.

The current technical leaders are all very busy with project work. The human resource manager actively advertises positions on both the internet and local newspapers which results in a large number of responses. It is difficult to sort through all the applicants as a high level of technical knowledge is required to vet them.

When the human resource (HR) manager selects a number of potential candidates from the vetted curriculum vitas, phone interviews are conducted. The HR manger pools questions which have been submitted from employees who have experience in the relevant technologies. The HR manager has to decide if the candidate's answers are satisfactory for the questions. The most successful candidates are then requested to attend a formal interview. Once a candidate presents for interview they are requested to take a short 10 minute written exam where they are asked to answer five questions relevant to the position they are applying for. Regardless of the outcome of the exam the candidate then proceeds to a formal interview where two or more employees from Devplex Technology interview the candidate and take note of their findings. Once the candidate has left the interview, the HR manager and interviewers meet to discuss the

exam and interview and decide if the candidate should be brought for a second interview. If the candidate's second interview is successful the candidate is hired.

Devplex Technology interviews a high number of unsuccessful candidates resulting in wasted time and effort. Sometimes employees who are not technically strong enough can be erroneously hired.

Devplex Technology wishes to reduce this workload and hire more suitable people by implementing an enterprise based candidate assessment system. The system should allow the remote assessment of potential candidates. It should also allow the HR manager to easily retrieve questions and answers on a selected topic. The system should test the candidates only on subjects which apply to the role they are hired for, the questions should progressively get harder as the candidate gets more questions correct, this will allow for a truly strong candidate achieve the highest score.

The overall aim of the system is to reduce workload and help find the best possible candidate for Devplex Technologies.

Acknowledgements

I would especially like to thank Mike Nims for advising me on this project and Regis & Galway Universities for this great educational opportunity.

Table of Contents

List of Figures

List of Abbreviations

ANT – Another Neat Tool

AJAX – Asynchronous JavaScript and XML

API – Application Programming Interface

ASP – Active Server Pages

CAS – Candidate Assessment System

CDAP – Common Display Architecture Processing

COS – Connection Oriented Service

CSS – Cascading Style Sheet

CV – Curriculum Vitae

D2D – Direct to DOM

DOM – Document Object Model

DTD – Data Type Definition

EBXML – Electronic Business Extensible Markup Language

EJB – Enterprise Java Bean

HTML – Hyper text markup language

HTTP – Hyper Text Transport Protocol

HTTPS – Hyper Text Transport Protocol Secure

HR – Human Resource

IDE – Integrated Development environment

IoC – Inversion of Control

ISV – Independent Service Vendor

J2EE – Java 2 Enterprise Edition

JAAS – Java Authentication and Authorization Service

JDO – Java Data Object

JNDI – Java Naming Directory Interface

JPA – Java Persistence API

JSF – Java Server Faces

JSP – Java Server Pages

NIS – Network Information Service

MS – Microsoft

MS IL – Microsoft Intermediary Language

MVC – Model View Controller

ORM – Object Relational Mapping

PC – Personal Computer

POJO – Plain Old Java Object

UI – User Interface

URL – Unified Resource Location

RAM – Random Access Memory

ROM – Read Only Memory

RDBMS – Relational Database Management System

SMTP – Simple Mail Transport Protocol

SOAP – Simple Object Access Protocol

SQL – Simple Query Language

W3C – World Wide Web Consortium

XML – Extensible Markup Language

XP – Extreme Programming

Chapter 1 – Introduction

Statement of the Problem

Devplex Technologies Limited is an expanding software services company. As with many software services companies staff turnover is high and there is always a constant stream of permanent and contract staff.

The human resource department needs a way of assessing potential candidates before formally interviewing them. Much time and effort is wasted interviewing unsuitable candidates.

Thesis Statement

Many HR departments spend a lot of time interviewing candidates for a particular position. The people working in HR and some managers may not be experienced in technical areas such as Java, Oracle and J2ee that the candidate is applying for. This means that a member of staff who is experienced in the area that the position is for will need to take time away from their work to conduct an interview to gauge the candidates knowledge in that particular area.

The solution to this problem is to build a web based electronic system that would ask the candidate questions based on the candidates skills profile.

The candidate would take an electronic multiple choice exam before the formal interview begins. The exam results are made available to the technical interviewer for a five minute review. Thereafter the formal interview would commence. Should the technical person have doubts then they will join the interview for the last 15 minutes only. This saves them sitting through the non-technical part of the interview.

Another area where the system will be helpful is for phone interviews. There will be technical questions available that non-technical interviewers may use thereby saving the time and effort of the technical staff.

When a candidate presents for an interview they will take an exam. Based on the candidate's CV the interviewer can define a candidate profile in the application. For example, if the candidate includes Sybase and Solaris experience in their CV then a Solaris and Sybase profile can be defined for the candidate.

When the candidate takes the exam the application will draw randomly from the pool of category questions relevant to the candidate's profile - in this example Sybase and Solaris.

If the candidate answers two successive easy questions correctly then they are asked a medium difficulty question; two successive correct answers of medium questions promotes the candidate to difficult questions. An incorrect answer demotes the candidate to the previous level until he/she gets two successive correct answers again. Points will be awarded as follows: 10 points for an easy, 20 points for medium and 30 points for difficult questions. With 18 questions the max score is 480. The exam will be time limited.

Project Objective

The objective of this project is to provide a workable prototype of a candidate assessment system.

It should meet the following objectives.

- Remote accessibility
- Secure
- Easy/Intuitive to use
- Low maintenance
- Low running costs

Remote accessibility is the ability to access the system from outside of the corporate network. This will mean that the system can be used by anyone using a HTML compliant internet browser on any device such as a laptop, palm pilot or mobile phone.

Secure: the system must be secure as candidate's details, test results and questions are maintained on the system. All users will be required to login. Candidates, administrators and HR personal should have certain system privileges which only allow them to perform specific actions.

Easy/Intuitive to use: the system should be user friendly. The user interface should follow proven current user interface styles. The exam user interfaces should be carefully constructed as to not waste the candidates time in trying to figure our how to use the assessment system instead of completing the exam.

Low Maintenance: Devplex Technology does not intend to spend a lot of time maintaining the system. Database backups should be automated. Also automatic recovery should be implemented in the event of a power failure. The only real maintenance should be the administration of users, their privileges and addition of exam questions for new subject areas.

Low Running Cost: there is no reason for the system to have high running costs. Very little maintenance will be required. It will also be possible to run the Database and Application server on a single server, reducing the number of servers required and the overall power consumption of the system.

Scope of the project

The scope of the project is to build and deploy a workable prototype of the candidate assessment system.

It system will include the following features:
Web interface – This type of interface provides the most portable, accessible and user friendly interface available. Most web interfaces are provided using HTML this allows a multitude of devices to access the content without any specific coding for such devices. The interface will allow the administrators to add, remove and edit candidates, questions and exams. The interface will also allow candidates to partake in exams.

Administration suite – This component of the system will allow administrators and HR mangers to add, remove and edit candidates, exam questions, exam answers and exams.

Candidate suite – This component of the system will allow candidates to take exams, iterate forward and backwards through the questions, change their answers and finally receive the results of their exam.

Multiple choice questions – The exam system will contain questions which can have a number of answers. It will be possible to select multiple answers for a single question

Single answer questions – Only one answer can be selected from a set of answers for this particular type of question.

Timed based exam – The exam will be limited by time. Once the time has expired the exam will be over.

Multi-user – It is important that many candidates can take the exam simultaneously. The system will be multi-user which will allow a number of candidates and/or administrators to use the system at the same time.

Challenges and Barriers

The following are considered the major challenges for this project.

Suitable deployment environment – The system will need to be remotely accessible. This includes outside of Devplex Technologies corporate firewall. The environment will need to be connected to the World Wide Web and support protocols such as HTTP, HTTPS and SMTP in order to provide end users with an accessible method of interacting with the system.

A database will be required in the environment to provide persistent storage for the exams, users and questions.

A domain name service will be required to provide an easy-to-remember URL for the system. Users will be able to access the site using the following URL through their internet browser www.devplex.ie. Without the DNS, the users would have to remember the IP address of the server. Using just an IP address as a method of access can lead to maintenance and convenience problems if the IP address of the server needs to be changed.

Mail server: the system will be able to send automated emails using a mail server over the STMP protocol. This will be very useful for the delivery of exam results or maintenance downtime alerts.

Application Server: this is essentially the heart of the environment. The application will run on the application server. The application server will be able to handle multiple requests giving concurrent user access. The application server will be used to handle the dynamic content.

Web Server: the web server will be the first point of contact when the user initiates a request. The web sever will handle all static content such as HTML pages and images, request for dynamic content will be passed onto the application server.

Presentation of exam user interface: this will be a particularly important challenge. The exam interface needs to be easy to use, responsive and accessible. It is envisaged that the interface will provide a desktop application like user experience where the web pages will not go blank as the content changes.

A steep learning curve of new technologies: the system will be constructed using cutting edge modern technologies. The development team will probably not have any previous experience in these technologies so it will take some time to learn the best development techniques with the various technologies that are being used.

Time: the system needs to be developed in a timely manner. As always estimating development time with unknown technologies can be difficult. Therefore estimates will be determined keeping this in mind.

Chapter 2 – Review of Literature and Research

Annotated Bibliography

Software Security: Building Security in, is a book written by Gary McGraw. It focuses on software security from inside the perimeter of the firewall as opposed to the traditional approach of relying on the network to provide much of the security needed. This book really brings home the point that the devil is in the details when it comes to application security. Overlooking a small detail can leave a gapping whole in the application which can expose vulnerabilities.

Gary McGraw. (2006) *Software Security: Building Security in.* Pearson Education, Inc.

“A central and critical aspect of the computer security problem is a software problem. Software defects with security ramifications—including implementation bugs such as buffer overflows and design flaws such as inconsistent error handling—promise to be with us for years. All too often malicious intruders can hack into systems by exploiting software defects. Moreover, Internet-enabled software applications are a commonly (and too easily) exploited target, with software's ever-increasing complexity and extensibility adding further fuel to the fire. By any measure, security holes in software are common, and the problem is growing.

The security of computer systems and networks has become increasingly limited by the quality and security of the software running on constituent machines. Internet-enabled software, especially custom applications that use the Web, are a sadly common target for attack. Security researchers and academics estimate that more than half of all vulnerabilities are due to buffer overruns, an embarrassingly elementary class of bugs. Of course, more complex problems, such as race conditions and design errors, wait in the wings for the demise of the buffer overflow. These more subtle (but equally dangerous) kinds of security problems appear to be just as prevalent as simple bugs.

Security holes in software are common. Over the last five years the problem has grown. Figure 1 shows the number of security-related software vulnerabilities reported to the CERT Coordination Center (CERT/CC) from 1995 through 2004. There is a clear and pressing need to change the way we approach computer security and to develop a disciplined approach to software security."

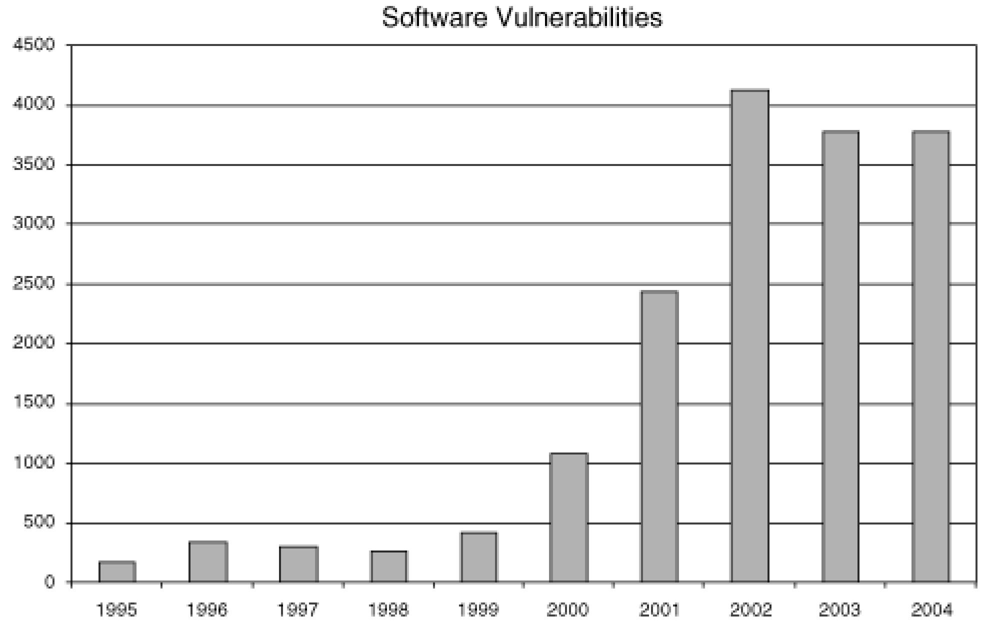


Figure 1 Software Vulnerabilities.

Designing Interfaces is a publication by Jenifer Twidwell, which describes the best practices to use when designing a user interface. It describes recommended approaches, best practices, design alternatives and warnings on where and when to use particular designs. The publication assumes some basic understanding of interface design in terms of information technology.

Jenifer Twidwell. (2005). *Designing Interfaces*. O'Reilly

"It's because good interface design doesn't start with pictures. It starts with an understanding of people: what they're like, why they use a given piece of software, and how they might interact with it. The more you know about them, and the more you empathize with them, the more effectively you can design for them. Software, after all, is merely a means to an end for the people who use it. The better you satisfy those ends, the happier those users will be.

Each time someone uses an application, or any digital product, they carry on a conversation with the machine. It may be literal, as with a command line or phone menu, or tacit, like the "conversation" an artist has with her paints and canvas—the give and take between the craftsperson and the thing being built. With social software, it may even be a conversation by proxy. Whatever the case, the user interface mediates that conversation, helping the user achieve whatever ends he or she had in mind.

As the user interface designer, then, you get to script that conversation, or at least define its terms. And if you're going to script a conversation, you should understand the human's side as well as possible. What are the user's motives and intentions? What "vocabulary" of words, icons, and gestures does the user expect to use? How can the application set expectations appropriately for the user? How do the user and the machine finally end up communicating meaning to each other?

There's a maxim in the field of interface design: "Know thy users, for they are not you!"”

Bulletproof Ajax is a recent publication by Jeremy Keith which constitutes an in-depth step by step guide on the best practices on how to enhance websites with Ajax functionality. The author uses progressive enhancement techniques to ensure graceful degradation. This will help the pages designed to work in all browsers. Cross browser portability is becoming increasingly important as more devices come on the market with internet ready technology. The reader is assumed to have a good working knowledge of Java Script, HTML and CSS.

Jeremy Keith. (2007). *Bulletproof Ajax.* New Riders

“Simplified definition of Ajax: a way of communicating with a Web server without refreshing the whole page.

This definition can provoke one of two responses. You could shrug your shoulders and say, "What's the big deal?" Alternatively, you could widen your eyes and exclaim, "That's amazing! It's a whole new paradigm for the Web!"

The truth about Ajax lies somewhere in between. It is an exciting technology. The ability to refresh just part of a Web page with information from a server can be used to great effect. On the other hand, Ajax is just a tool. By itself, it can't create a good user experience. Content is still king.
Many technologies can fall under the umbrella of Ajax.

Adobe Flash movies are now capable of communicating asynchronously with a Web server. That means that the contents of a Flash movie can be updated without a page refresh. That sounds like Ajax.

The Adobe Flex framework has given developers even more power. Flash is now a viable technology for delivering rich, responsive Web applications.

Java applets are little programs written in Java, not to be confused with JavaScript. Much like Flash movies, these programs can be embedded in Web pages. They are also capable of communicating with the server even after they have loaded.
The speed and responsiveness of the applets varies enormously depending on the specifications of the end user's machine. Java applets have never really taken off.

Frames aren't used very much these days, mostly because they're a usability nightmare. If a Web page is built using a frameset, just one frame can be updated without updating every frame in the page. Using an inline frame, or iframe, is a step up from a frameset. An iframe can also be used as a secret conduit to a Web server. If a Web page contains a tiny, practically invisible iframe, its source can be constantly updated. Using JavaScript, the parent page can gather information from the updated iframe. Google Maps uses a hidden iframe to communicate with the server. It's a clever solution, although it does feel slightly hackish.

The XMLHttpRequest object is an extension to JavaScript that allows Web pages to communicate with a server. It's perfect for creating Ajax applications. Jesse James Garrett had XMLHttpRequest in mind when he coined the term Ajax.
The biggest problem with XMLHttpRequest is how long it takes to say it. Even though there is an X in it, it was never going to catch on as a buzzword. The word Ajax is a lot shorter and snappier, and it's usually synonymous with using the XMLHttpRequest object."

Microsoft .NET vs. J2EE: How Do They Stack Up? is a publication by Jim Farley. The pages constitute an overview of J2EE (Java 2 Enterprise Edition) and Microsoft's .NET technology. It then summarizes and compares both technologies in a table format. The article assumes some general technical knowledge on the part of the reader particularly on the J2EE side.

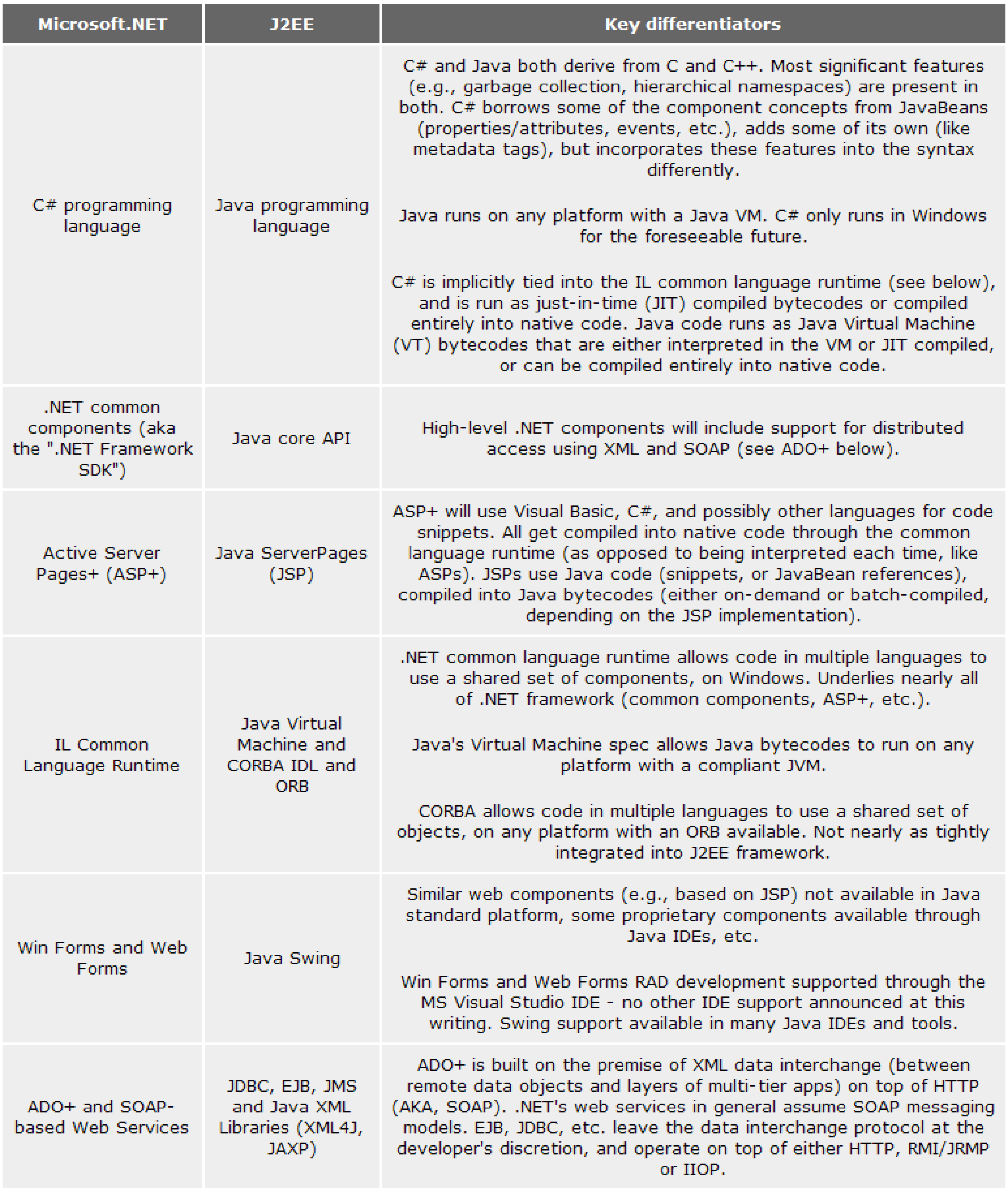
Jim Farley. (2001). *Microsoft .NET vs. J2EE: How Do They Stack Up?.* Retrieved July 18th 2007 from the O'Reilly publication website: http://www.oreillynet.com/pub/a/oreilly/java/news/farley_0800.html

"It would be easy to dismiss .NET as more Microsoft marketing-ware and continue on your merry way. But don't. .NET is a sign of a subtle but significant shift in Microsoft's strategy to evangelize their platforms. They have been fighting alternative frameworks and platforms at the management level pretty well, touting the usual questionable "statistics" about cost of ownership and seamless integration. Now they are fighting Java and open source initiatives on their own terms, putting their own spin on "open" and attempting to directly address the needs of developers, two things that they have been faulted for not doing very well in the past. If you consider yourself an evangelist for Java or open source platforms, then the nature of the war is changing. Be prepared.

Also, Microsoft's IL runtime has at least one notable, if improbable, goal: eliminate the programming language as a barrier to entry to the framework. Java eliminates the platform barrier (within limits, of course: You can't make up for missing hardware resources with software, for example), but in order to work in J2EE, you have to work in Java. .NET wants to let you use the language of your choice to build .NET applications. This is admirable, though there are big questions as to whether and when the IL approach in .NET will actually become broadly useful. Regardless, this points to a weakness in the single-language J2EE approach. The importance of this weakness is questionable, but it exists nonetheless, and deserves some consideration by the Java community. If this is really desired by developers, then maybe the efforts in Java bytecode generators for non- Java languages should be organized and consolidated.

Focusing on J2EE, there are a few issues that should be addressed immediately in order to bolster the advantages of that platform compared to what .NET is shooting for. First, XML support needs to be integrated seamlessly into the framework. I'm not talking about bolting an XML SAX/DOM parser to the set of standard services, or extending the use of XML in configuration files. XML messaging and manipulation need to be there, ready to use. Admittedly, you can use XML payloads on top of JMS messaging, but the platform doesn't facilitate this at all. The XML space is a cluttered mess of standards, de facto standards, APIs and DTDs, which is to be expected when you're dealing with a meta-language.

But Microsoft has put a stake in the ground with SOAP, and they're pushing hard to put something understandable and useful in the hands of developers. J2EE proponents need to do the same with their platform. One possibility that comes to mind is to add an XML messaging "provider" layer on top of JMS, along the lines of the pattern followed by Java Naming and Directory Interface, or JNDI, with LDAP, NIS, COS Naming, etc. This in combination with a standard SOAP/BizTalk provider, an ebXML provider, etc. would be an impressive statement."



Microsoft.NET	J2EE	Key differentiators
C# programming language	Java programming language	C# and Java both derive from C and C++. Most significant features (e.g., garbage collection, hierarchical namespaces) are present in both. C# borrows some of the component concepts from JavaBeans (properties/attributes, events, etc.), adds some of its own (like metadata tags), but incorporates these features into the syntax differently. Java runs on any platform with a Java VM. C# only runs in Windows for the foreseeable future. C# is implicitly tied into the IL common language runtime (see below), and is run as just-in-time (JIT) compiled bytecodes or compiled entirely into native code. Java code runs as Java Virtual Machine (VT) bytecodes that are either interpreted in the VM or JIT compiled, or can be compiled entirely into native code.
.NET common components (aka the ".NET Framework SDK")	Java core API	High-level .NET components will include support for distributed access using XML and SOAP (see ADO+ below).
Active Server Pages+ (ASP+)	Java ServerPages (JSP)	ASP+ will use Visual Basic, C#, and possibly other languages for code snippets. All get compiled into native code through the common language runtime (as opposed to being interpreted each time, like ASPs). JSPs use Java code (snippets, or JavaBean references), compiled into Java bytecodes (either on-demand or batch-compiled, depending on the JSP implementation).
IL Common Language Runtime	Java Virtual Machine and CORBA IDL and ORB	.NET common language runtime allows code in multiple languages to use a shared set of components, on Windows. Underlies nearly all of .NET framework (common components, ASP+, etc.). Java's Virtual Machine spec allows Java bytecodes to run on any platform with a compliant JVM. CORBA allows code in multiple languages to use a shared set of objects, on any platform with an ORB available. Not nearly as tightly integrated into J2EE framework.
Win Forms and Web Forms	Java Swing	Similar web components (e.g., based on JSP) not available in Java standard platform, some proprietary components available through Java IDEs, etc. Win Forms and Web Forms RAD development supported through the MS Visual Studio IDE - no other IDE support announced at this writing. Swing support available in many Java IDEs and tools.
ADO+ and SOAP-based Web Services	JDBC, EJB, JMS and Java XML Libraries (XML4J, JAXP)	ADO+ is built on the premise of XML data interchange (between remote data objects and layers of multi-tier apps) on top of HTTP (AKA, SOAP). .NET's web services in general assume SOAP messaging models. EJB, JDBC, etc. leave the data interchange protocol at the developer's discretion, and operate on top of either HTTP, RMI/JRMP or IIOP.

Figure 2 .NET vs J2EE

The Java Programming Language, an article by David Reilly where he discusses the revolutionary programming language called Java and offers a glimpse behind the Java platform. The following core aspects of the language are introduced: Object-oriented, Portable, Multi-threaded, Automatic garbage collection, Security, Network & internet awareness and Ease of use.

David Reilly. (n.d.). *The Java Programming Language.* Retrieved July 18 2007, from http://www.javacoffeebreak.com/articles/inside_java/insidejava-nov99.html

"The properties that make Java so attractive are present in other programming languages. Many languages are ideally suited for certain types of applications, even more so than Java. But Java brings all these properties together, in one language. This is a revolutionary jump forward for the software industry."

J2ee Design Patterns is a publication by William Crawford and Jonathon Kaplan. The book comprises of a catalogue of Design Patterns related to J2ee development. But it is not just simply a catalogue of design patterns; the authors broaden the scope by discussing when, where and why the design pattern is suited to a particular problem. It is assumed that the reader has at least an elementary understanding of J2ee and a good knowledge of Java.

William Crawford and Jonathon Kaplan. (2003). *J2EE Design Patterns.* O'Reilly

"A J2EE environment differs from a Java 2, Standard Edition (J2SE) environment in that it offers a wider range of services than a standalone application could expect to find. J2SE is geared towards providing core language features (I/O, text manipulation, variables, object semantics), standard utility classes that apply in a variety of settings (collections, mathematics), and features required for building client applications (GUIs, and some basic enterprise integration, including access to databases and naming services).

The J2EE application model is built on a division of labor into various tiers. The client presentation in a web browser, applet, or Java application is separated from server side logic in a JavaServer Page or Java Servlet and the business logic in a database or Enterprise JavaBeans. The J2EE APIs are focused on implementing the interactions between tiers. The interfaces to each tier are standardized, allowing programmers with an understanding of the core J2EE concepts to easily apply their skills to any J2EE-based project.

The core of a J2EE application deployment is a J2EE-compliant application server. The application server supports hosting business logic components and web components, providing access to the full range of J2EE capabilities. Note that the J2EE API doesn't say very much beyond the essentials about how these servers should be designed, or how deployment, maintenance, and general administration should be conducted. The focus of the J2EE API, instead, is on programmatic interfaces and runtime behavior. This specialization can make it difficult to transfer administration skills from, say, IBM WebSphere to BEA WebLogic. Code, however, should transfer transparently. Figure 3 shows the canonical J2EE application model.

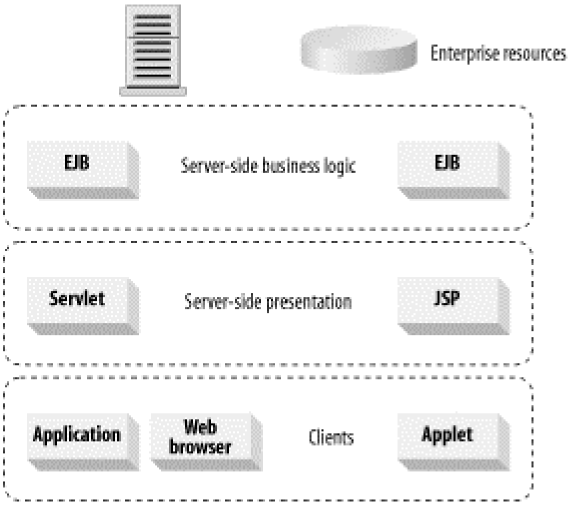


Figure 3 Canonical J2EE application model

Each J2EE API itself is simply a wrapper for a service provided by the J2EE container or by an external component within the enterprise. A full-scale J2EE environment could include one or more J2EE application servers hosting servlets and Enterprise JavaBeans, as well as an external transaction manager, an Oracle or DB2 relational database, and a messaging middleware system.

Most web application frameworks implement at least the MVC pattern, and many even claim to be "pattern-based." In general, frameworks force applications written to them to conform to certain patterns. Recognizing these patterns can be a big help in designing applications that work well in a given framework. Here's a look at two popular frameworks and the patterns they implement:

Jakarta Struts is probably the best-known web application framework. It is based on what they call a "Model 2" approach, an implementation of MVC in which the model and view components can be written and Struts provides the controller. The Struts controller is based on the Service to Worker pattern, featuring a front controller, a dispatcher configured via XML files, and a code framework for embedding application logic into actions. Struts also provides a number of utility classes and tag libraries, including Tiles, an implementation of the Composite View pattern.

JavaServer Faces (JSF) is a new web application framework from Sun. The primary goal of JSF is to provide a UI framework for web applications, allowing the development of custom, reusable UI widgets that maintain state across requests. As such, JSF is centered around the Composite View pattern, and includes an extensive tag library for developing and using these components. JSF also provides a simple implementation of the Service to Worker pattern, including a front controller, an XML-configured navigation handler, and support for actions.

Jakarta Struts Cookbook is a publication by Bill Siggelkow on the widely used J2EE framework Jakarta Struts. The book constitutes a large collection of solutions to common and uncommon problems encountered when implementing an application with the struts framework. Topics such as security, processing, internationalization, etc are all covered in detail. The book also details which types of applications Struts is best suited to. The book assumes that the reader has some experience with Struts. It's expected that the reader know how to build and deploy a J2EE web application.

Bill Siggelkow. (2005). *Jakarta Struts Cookbook*. O'Reilly

"Face it: you can create the most architecturally pure, elegant, and robust web application, but if the users don't like the looks of the interface, you are doomed to failure. Some Java developers consider themselves above the use of such mundane technologies as HTML and JavaScript. Whether you like it or not, however, knowledge of these technologies—particularly HTML—can make all the difference when it comes to presentation and usability. If you don't know how to use them to your advantage, you will find it challenging for your application to be endorsed by the user community.

Any application can shine under normal conditions; how an application responds to unexpected conditions reveals much more about its robustness and production-readiness. The Struts Validator provides a means for handling unexpected user input. However, applications need to handle unexpected system behavior. This behavior typically manifests as exceptions. These exceptions may be business-related, may have meaning to the end user, or they may be system- and coding-related, or may have meaning to system administrators and developers.

The heart of testing and debugging is the unit test. A unit test verifies that a specific portion of your application does what it is supposed to do. Java developers commonly think of the "unit" as a Java class. But the unit can be anything as long as it can be identified by discrete boundaries and isolated from code external to the unit.

In "traditional" software testing, the tester ensures that the application functions as the user expects. You can apply the concepts of unit testing to functional testing by considering a scenario as the "unit." Use cases such as "Login," "New User Registration," or "Place an Order" to drive your unit tests. These functional unit tests are as valid as the low-level tests, though these tests can be more difficult to isolate from outside effects. If you follow the best practices of web application design—using the Model-View-Controller (MVC) design pattern supported by Struts—your most important unit tests will be testing your application's model and business services. Struts-specific tests should comprise a small portion of your application's entire unit test suite.”

Spring 2.0: What's New and Why it Matters, is an article posted by Rod Johnson. This article gives a brief overview of Spring and looks at some of the following new features in more detail, illustrating them with code examples.

- XML Configuration Extensions
- Out-of-the-box-namespaces
- Third-party Configuration Extensions
- Syntax Sugar
- New Bean Scopes
- Web Scopes
- Type inference
- New Extension Points
- AOP Enhancements

The reader is assumed to have knowledge of the previous version of Spring, some knowledge of XML and Java.

Rod Johnson. (2007). *Spring 2.0: What's New and Why it Matters.* Retrieved July 18th, from http://www.infoq.com/articles/spring-2-intro

“Since the open source project began in February, 2003, the Spring Framework has gone from strength to strength. It has powered past 1 million downloads; become a de facto standard in a wide range of industries; and changed the development of enterprise Java applications.

Most important, it has developed a large and loyal user base, which understands its key values and has contributed feedback that has helped it to advance rapidly. Spring's mission has always been clear:

- To provide a non-invasive programming model. As far as possible, application code should be decoupled from the framework.
- To provide a superior solution to in-house infrastructure, so that developers can focus on delivering business value rather than solving generic problems.
- To make developing enterprise applications as simple as possible, but enhancing, rather than sacrificing, power.”

Hibernate simplifies inheritance mapping

Hibernate is an object relational mapping tool which uses XML configuration files to map tables in a relational database to plain old Java objects. This article is introductory level but does assume a strong knowledge of Java, Relational Databases and XML. A number of different strategies are explained to implement inheritance using Hibernate.

Xavier Coulon and Christian Brousseau. (2004). *Hibernate simplifies inheritance mapping*. Retrieved July 18th, from http://www.ibm.com/developerworks/java/library/j-hibernate/

“Hibernate is an object-relational mapping and persistence framework that provides a lot of advanced features, ranging from introspection to polymorphism and inheritance mapping. But mapping class hierarchies to a relational database model might prove somewhat difficult. This article covers three strategies that you can use in your everyday programming to easily map complex object models to relational database models."

Real World Experiences with Hibernate

This article is an overview of some of the lessons learned by a company called Shine Technologies during development with Hibernate.

The article does not explain how to use Hibernate nor does it consider the examples given as best practice but it goes into detail on testing, debugging, performance, tools, annotations and deployment.

Shine Technologies. (n.d.). *Real World Experiences with Hibernate.* Retrieved July 21st, from http://shinetech.com/pages/viewpage.action?pageId=649

"When it comes to using Hibernate, in our experience there's both good news and bad news. The bad news is that it's a big piece of software and can be difficult to use - probably more so than you realize. The good news is that it is only as hard as you want it to be.

Generally, the more of it you choose to use, the more time it can save you. However, the flipside of this is that the more you use it, the more it will pervade your architecture and thus the greater the risk that it will leave you tangled up in a big knot if you don't fully understand what you're doing.

Another thing worth knowing about Hibernate is that there are many ways to skin a cat. In other words, there can be many means to accomplishing what looks like the same end. The important thing to realize is that you will often be trading-off a number of factors - whether you know it or not.

So what are these tradeoffs? Well, here are a few important ones:

- Correctness. This may seem obvious, but a seemingly innocent change can cause a breakage if you're not careful. This is especially the case with domain objects and their mappings. You've got to make sure that in modifying a domain object to fix a problem for one of its clients, you don't introduce problems for all the other clients. For example, changing the fetch strategy of a relationship in a domain object may work fine if the object is attached to a session, but may not work if the object is detached.

- Performance. If you aren't careful, what may seem like a minor change can actually result in your application no longer reaching its performance targets.

- Conciseness. In other words, the number of lines of code that your solution occupies. A common trap that people fall into is to try and solve a problem using an advanced feature of Hibernate. Whilst this in itself is not bad, the problem is that it can affect...

- Development Time. When it takes two days to figure out how to do something instead of the two hours it'd take if you'd used a simpler approach, you may have a problem!

In light of all this, the chief advice we would give you in your dealings with Hibernate is to think critically about each of decisions that you make. Do not blindly accept "best practices", be they from the Hibernate team or anybody else. Instead, aim to keep things as simple as possible. Admittedly, in non-trivial systems the simplest path is not always obvious. However, it still a goal worth aiming for.”

Jumping into JBoss

JBoss is a free, open source, application server that implements the complete Java 2 Enterprise Edition (J2EE) stack, including Java Server Pages (JSP), servlets, and Enterprise JavaBeans (EJB). For J2EE developers that are getting started with JBoss, this article presents the basics, including downloading, installation, application deployment, and data source configuration.

This article is designed to get a developer up and running quickly with JBoss, it covers everything from installation to deploying an example application.

Thornton Rose and David Thurmond. (n.d.). *Jumping into JBoss.* Retrieved June 1st, from http://www.developer.com/java/ejb/article.php/3071661

"JBoss comes with three server configurations:

- minimal - This configuration includes only logging, the JNDI service, and the URL deployment scanner. You would want to use this configuration for starting services that don't require J2EE or as the base for a custom configuration.
- default - This configuration is the default. It includes all of the J2EE services exception RMI/IIOP and clustering.
- all - This configuration includes all JBoss services."

Developing Web Applications with JavaServer Faces

In this article Qusay H. Mahmoud, gives a detailed explaination of what JSF is. He first provides an overview of the core concepts and how JSF differs from standard web technology, then a detailed example of a simple JSF application is used to present some of the finer points. It is assumed that the reader has a good knowledge of Java and Web applications in general.

Qusay H. Mahmoud. (2004). *Developing Web Applications with JavaServer Faces.* Retrieved June 12th, from http://java.sun.com/developer/technicalArticles/GUI/JavaServerFaces/

"JavaServer Faces (JSF) is a standardized specification for building User Interfaces (UI) for server-side applications. Before JavaServer Faces, developers who built web applications often relied on building HTML user interface components with servlets or JavaServer Pages (JSP pages). This is mainly because HTML user interface components are the lowest common denominator that web browsers support. The implication, of course, is that such web applications do not have rich user interfaces, compared with standalone fat clients, and therefore less functionality and/or poor usability. While applets can be used to develop rich user interfaces, web application developers don't always know what clients will be accessing the application and/or they may have no access to the client device.

In addition, if you have taken part in developing a large-scale web system, you may have come across technical challenges, such as how to implement custom components like a query builder or a table viewer for building database queries. Building such custom components requires expertise and a significant amount of time to build and test the new libraries. In an ideal environment, developers would be able to use prebuilt, tested, and highly configurable components to integrate into their application development environment.

JavaServer Faces (JSF) is a server-side technology for developing web applications with rich user interfaces. With JSF, you can resolve such technical challenges as creating custom user interface components. This is because JSF technology is a user interface framework for building Java-based web applications that run on the server side, and render the user interface back to the client. That's right! The user interface code runs on the server, responding to events generated on the client." The component stack of JSF is depicted in Figure 4.

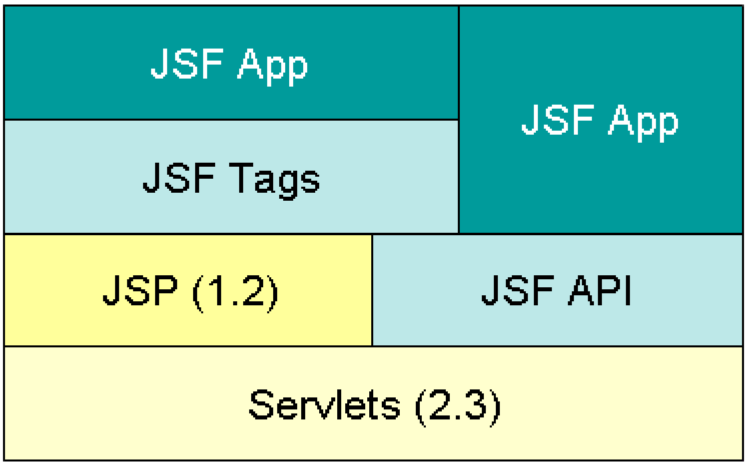


Figure 4 High level Architecture of JSF

ICEfaces Offers a Novel, Pure Java Approach to the Rich vs. Thin Dilemma.

JSF (Java Server Faces) is a relatively new framework for building web applications. It simplifies development by providing a component centric approach to developing user interfaces.

ICEFaces builds on top of the JSF framework by providing a means of updating the HTML (Hyper Text Markup Language) via AJAX (Asynchronous JavaScript and XML). The beauty of this approach means that individual HTML components can be updated without reloading the complete web page. This gives a far superior user experience to normal JSF.

The article gives a detailed introduction to ICEfaces and an explanation on how the technology functions.

Laurance Moroney. (2005). *ICEfaces Offers a Novel, Pure Java Approach to the Rich vs. Thin Dilemma.* Retrieved July 2nd, from http://www.devx.com/webdev/Article/29068

"Java Server Faces is used to provide a centralized, server-based model that preserves richness at the client. Sounds perfect, right? Almost. This model depends on a markup-based presentation and the existing Web application model, thus making it prone to the same limitations of Web-based apps, including development and maintenance difficulties.

ICEfaces, from ICEsoft Technologies attempt to improve this model—harnessing the Asynchronous Java and XML (AJAX) methodology to JSF in order to create a client-light, high-performance, rich user experience. For those who haven't already heard of them, ICEsoft is a provider of Web applications and toolkits for Java developers, having previously released ICEbrowser, a 100 percent pure Java Web browser; ICEreader, an HTML rendering engine for Java developers; and ICEpdf, a PDF rendering engine for Java developers.

What makes ICEfaces different is its Direct-to-DOM rendering, which builds on the JSF renderkit architecture and provides a separation between JSF components and the markup that represents them in the presentation layer. In a regular JSF application, a component is defined in the JSF component tree and the renderkit generates the appropriate markup for that component on that specific client; in other words, it produces a different result for a standard (HTML) Web browser or for a WML-based mobile device.

In Direct-to-DOM rendering a JSF component tree is rendered directly into a W3C DOM data structure. During the JSF render pass, this tree is traversed, and the appropriate output is generated for each component on the server. The changes to the DOM that result are packaged up and delivered to the browser via an AJAX bridge and used to create the presentation for the application (see Figure 5)."

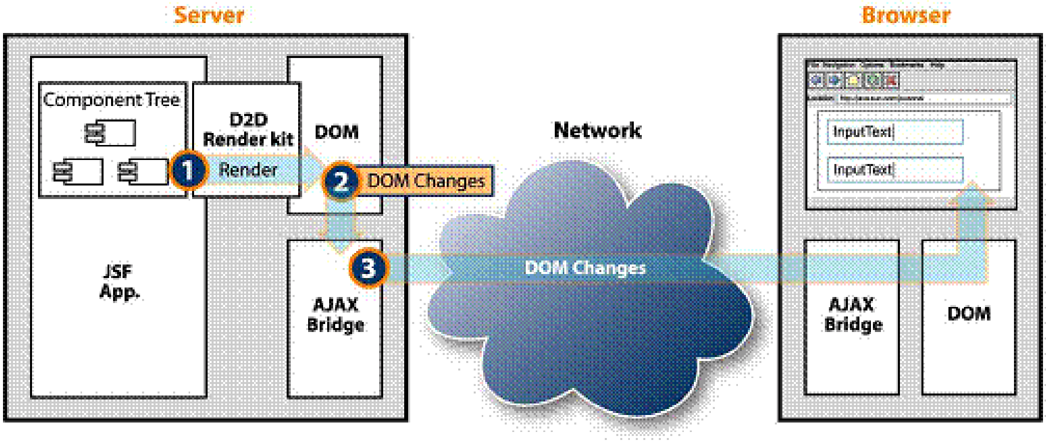


Figure 5 Direct-to-DOM Model

JAAS Security in Action

In this article Kyle Gabhart introduces the Java Authentication and Authorization Service (JAAS). He provides in-depth technical details on JAAS and supports these with a brief example. According to Sun's Web site, "JAAS is a set of packages that enables services to authenticate and enforce access controls upon users. It implements a Java version of the standard Pluggable Authentication Module (PAM) framework, and supports user-based authorization."

The article assumes a good understanding of JAVA technology in general.

Kyle Gabhart. (2002). *JAAS Security in Action.* Retrieved July 3rd, from http://www.devx.com/getHelpOn/Article/9915/0

“More than ever before, corporations are clamoring to evaluate their security architectures and identify any gaps. The Java platform, and specifically the J2EE platform, provides some of the most robust application-level security available today. The Java Authentication and Authorization Service (JAAS), which was introduced as an optional security package for the Java 2 SDK, Standard Edition, version 1.3, has been formally included as a part of the standard Java packages as of version 1.4.

JAAS authentication is deployed in a pluggable manner, using code modules that implement certain interfaces. This enables Java applications to remain decoupled from the underlying authentication technologies. Additional authentication protocols and updated authentication technologies can be plugged in at runtime without modifying the application or recompiling the source code.

The JAAS Authentication API is quite extensive. The key interfaces and classes that need to be familiarized are as follows:

- **Callback** – Implementations of this interface encapsulate information (usernames, passwords, error and warning messages) that is exchanged between security services and a CallbackHandler.
- **CallbackHandler** – An application implements a CallbackHandler and passes it to underlying security services to facilitate interaction between the security services and the application.
- **LoginContext** – The LoginContext class provides the basic methods used to authenticate Subjects in a neutral manner, decoupled from the underlying authentication technology.
- **LoginModule** – Authentication technology providers implement this interface to provide a particular type of authentication via a pluggable module.
- **Principal** – The Principal interface represents the abstract notion of a principal, which can be used to represent any unique entity (individual, corporation, organization, login id, social security number, etc.) that can be authenticated.
- **Subject** – A Subject object represents a grouping of related information for a single entity, such as a person. One or more Principals are bound to a subject. Each Principal represents one identity for the subject (name, social security #, etc.). A Subject also maintains security-related attributes (passwords and cryptographic keys, for example)."

Put JSF to Work

This article addresses the issues of integrating JSF (Java Server Faces) with some of the well proven frameworks used in J2EE at the moment specifically, the Spring Framework and Hibernate. It demonstrates how to create a typical web application, an online product catalog system. Using the example, this article covers each phase of web application design, including business-requirement gathering, analysis, technology selection, high-level architecture, and implementation-level design. The article discusses the advantages and disadvantages of the technologies used in the example and demonstrate approaches for designing some of the application's key aspects.

This article is written for Java architects and developers already working with J2EE-based Web applications. It is not an introduction to JSF, the Spring Framework or Hibernate.

Derek Yang Shen. (2004). *Put JSF to work.* Retrieved July 4th, from http://www.javaworld.com/javaworld/jw-07-2004/jw-0719-jsf.html

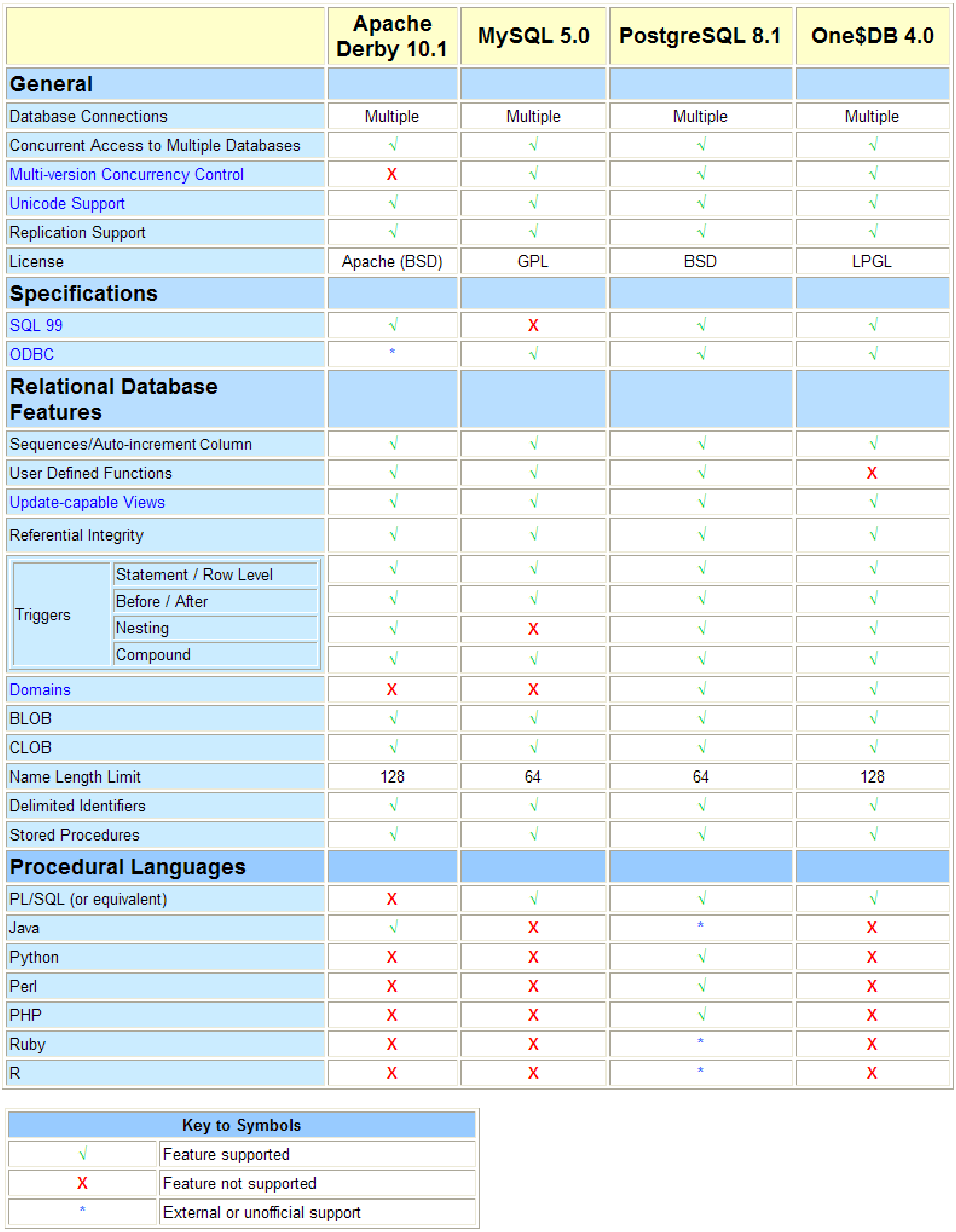
"Java Server Faces (JSF) technology is a new user interface framework for J2EE applications. It is particularly suited, by design, for use with applications based on the MVC (Model-View-Controller) architecture. Numerous articles have introduced JSF. However, most take a highly theoretical approach that doesn't meet the challenges of real-world enterprise development. Many issues remain unsolved. For example, how does JSF fit into the overall MVC architecture? How does JSF integrate with other Java frameworks? Should business logic exist in the JSF backing beans? How do you handle security in JSF? And most importantly, how do you build a real-world Web application using JSF?"

Open Source Database Feature Comparison Matrix

In this comparison matrix, the important features of the most predominant open source database are laid out in black and white. This makes it easy to see which databases support the features your project requires. Some understanding of database terminology is required.

Joshua D. Drake. (2005). *Open Source Database Feature Comparison Matrix.* Retrieved July 10th, from http://www.devx.com/dbzone/Article/29480

"Looking for a database? This handy database feature comparison matrix and glossary let you compare and contrast the supported features of four popular open source databases: Apache Derby 10.1, MySQL 5.0, PostgreSQL 8.1, and One$DB 4.0."



		Apache Derby 10.1	MySQL 5.0	PostgreSQL 8.1	One$DB 4.0
General					
Database Connections		Multiple	Multiple	Multiple	Multiple
Concurrent Access to Multiple Databases		√	√	√	√
Multi-version Concurrency Control		X	√	√	√
Unicode Support		√	√	√	√
Replication Support		√	√	√	√
License		Apache (BSD)	GPL	BSD	LPGL
Specifications					
SQL 99		√	X	√	√
ODBC		*	√	√	√
Relational Database Features					
Sequences/Auto-increment Column		√	√	√	√
User Defined Functions		√	√	√	X
Update-capable Views		√	√	√	√
Referential Integrity		√	√	√	√
Triggers	Statement / Row Level	√	√	√	√
	Before / After	√	√	√	√
	Nesting	√	X	√	√
	Compound	√	√	√	√
Domains		X	X	√	√
BLOB		√	√	√	√
CLOB		√	√	√	√
Name Length Limit		128	64	64	128
Delimited Identifiers		√	√	√	√
Stored Procedures		√	√	√	√
Procedural Languages					
PL/SQL (or equivalent)		X	√	√	√
Java		√	X	*	X
Python		X	X	√	X
Perl		X	X	√	X
PHP		X	X	√	X
Ruby		X	X	*	X
R		X	X	*	X

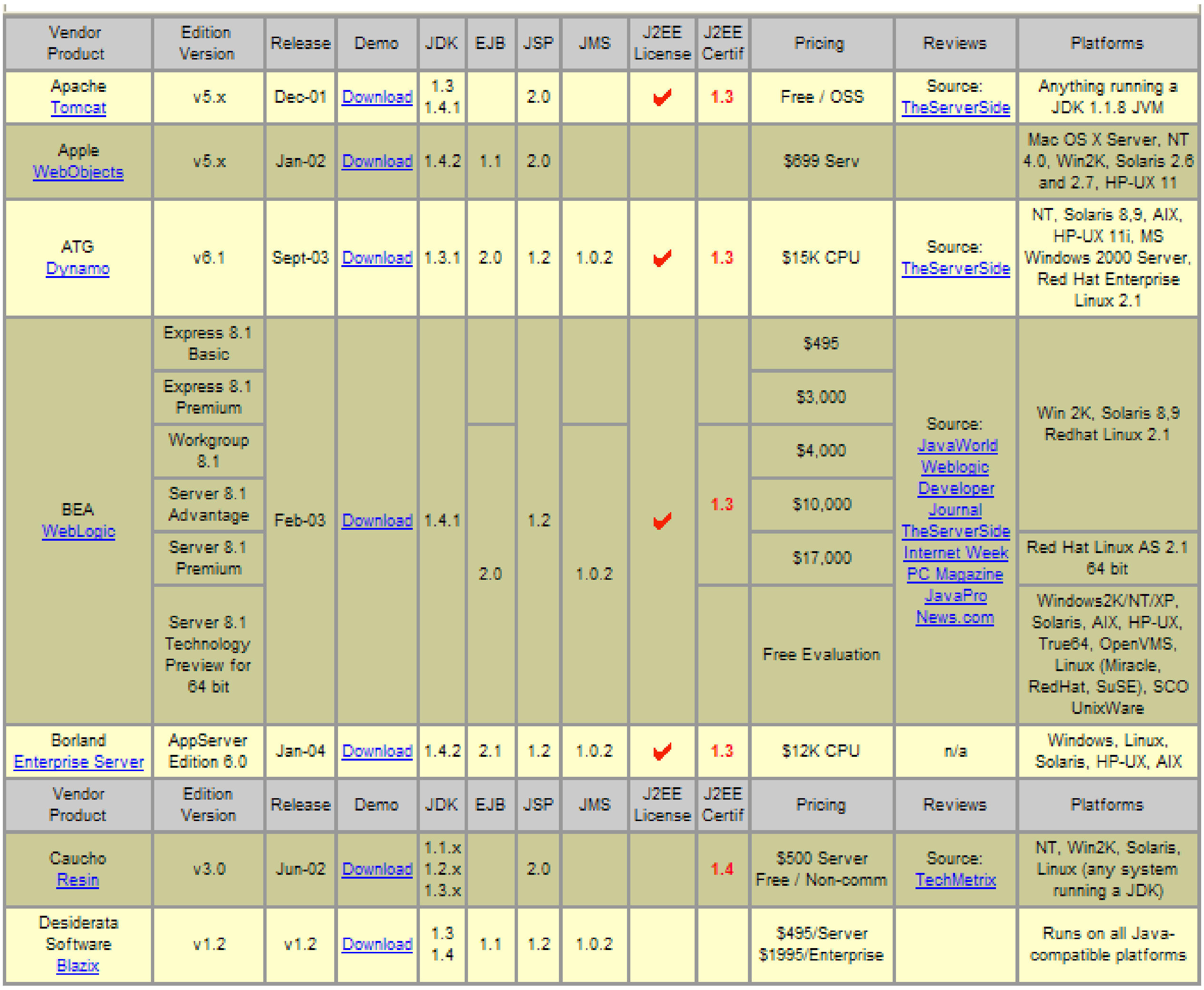
Key to Symbols	
√	Feature supported
X	Feature not supported
*	External or unofficial support

Figure 6 Partial Diagram – database matrix

Application Server Matrix

The matrix provides an overview of the features that each of the main Severs available provide.

The Server Side. (n.d.) *Application Server Matrix*, Retrieved July 20th, from http://www.theserverside.com/tt/articles/article.tss?l=ServerMatrix



Vendor Product	Edition Version	Release	Demo	JDK	EJB	JSP	JMS	J2EE License	J2EE Certif	Pricing	Reviews	Platforms
Apache Tomcat	v5.x	Dec-01	Download	1.3 1.4.1		2.0		✓	1.3	Free / OSS	Source: TheServerSide	Anything running a JDK 1.1.8 JVM
Apple WebObjects	v5.x	Jan-02	Download	1.4.2	1.1	2.0				$699 Serv		Mac OS X Server, NT 4.0, Win2K, Solaris 2.6 and 2.7, HP-UX 11
ATG Dynamo	v6.1	Sept-03	Download	1.3.1	2.0	1.2	1.0.2	✓	1.3	$15K CPU	Source: TheServerSide	NT, Solaris 8,9, AIX, HP-UX 11i, MS Windows 2000 Server, Red Hat Enterprise Linux 2.1
BEA WebLogic	Express 8.1 Basic	Feb-03	Download	1.4.1		1.2		✓		$495	Source: JavaWorld Weblogic Developer Journal TheServerSide Internet Week PC Magazine JavaPro News.com	Win 2K, Solaris 8,9 Redhat Linux 2.1
	Express 8.1 Premium									$3,000		
	Workgroup 8.1				2.0		1.0.2		1.3	$4,000		
	Server 8.1 Advantage									$10,000		
	Server 8.1 Premium									$17,000		Red Hat Linux AS 2.1 64 bit
	Server 8.1 Technology Preview for 64 bit									Free Evaluation		Windows2K/NT/XP, Solaris, AIX, HP-UX, True64, OpenVMS, Linux (Miracle, RedHat, SuSE), SCO UnixWare
Borland Enterprise Server	AppServer Edition 6.0	Jan-04	Download	1.4.2	2.1	1.2	1.0.2	✓	1.3	$12K CPU	n/a	Windows, Linux, Solaris, HP-UX, AIX
Vendor Product	Edition Version	Release	Demo	JDK	EJB	JSP	JMS	J2EE License	J2EE Certif	Pricing	Reviews	Platforms
Caucho Resin	v3.0	Jun-02	Download	1.1.x 1.2.x 1.3.x		2.0			1.4	$500 Server Free / Non-comm	Source: TechMetrix	NT, Win2K, Solaris, Linux (any system running a JDK)
Desiderata Software Blazix	v1.2	v1.2	Download	1.3 1.4	1.1	1.2	1.0.2			$495/Server $1995/Enterprise		Runs on all Java-compatible platforms

Figure 7 Partial diagram – server matrix.

JBoss AS 4.0

This article provides a review of the JBoss 4.0 Application server. It discusses some of changes and carryovers of the server.

Jim Rapoza. (2004). *JBoss AS 4.0.* Retrieved July 21st, from http://www.eweek.com/article2/0,1759,1668400,00.asp

“JBoss' updated Java application server offers the best of both the open-source and commercial software worlds. Version 4.0 has the excellent standards and platform support, community involvement and ease of integration typical of the open-source world, and the strong service, support and documentation common in the commercial world. With improved scalability and integration features and full support for J2EE 1.4, the free JBoss AS 4.0 is a good application server choice for enterprises of all sizes

Many companies choose the JBoss application server because of its extreme ease of deployment (you basically just unpack it and run it) and because it runs on almost every operating system platform out there. ISVs like it because of its strong support for standards, and its modular design makes it ideal for embedding within other applications.”

Extreme Programming: A gentle introduction is a publication by J Donovan Wells, which gives a complete overview of the extreme programming methodology from getting started through to deployment. The author gives advice on the best way to start using extreme programming weather to add a little to a current methodology or go for the big bang approach. Links to other resources are also provided.

J. Donovan Wells. (n.d.). *Extreme Programming: A Gentle Introduction.* Retrieved August 3rd 2007 from http://www.extremeprogramming.org/

"Let's begin with a simple question:

What is XP? As you will see, it is a deliberate and disciplined approach to software development.

Next we might wonder when to use XP. Risky projects with dynamic requirements are perfect for XP. These projects will experience greater success and developer productivity.

But do we need yet another software methodology? Actually we do. XP is a refreshing new approach. XP is successful because it emphasizes customer involvement and promotes team work.

So how could this possibly work? The most surprising aspect of XP is its simple rules and practices. They seem awkward and perhaps even naive at first, but soon become a welcome change. Customers enjoy being partners in the software process and developers actively contribute regardless of experience level."

What is Extreme Programming, an article by Ronald E Jeffries gives a quick introduction of extreme programming. He then goes into a very detail explanation of each of the core practices: Whole Team, Planning Game, Small Releases, Customer Tests, Simple Design, Pair Programming, Test-Driven Development, Design Improvement, Continuous Integration, Collective Code Ownership, Coding Standard, Metaphor and Sustainable Pace.

Ronald E Jeffries (2001). *What is Extreme Programming,* Retrieved July 21st, from http://www.xprogramming.com/xpmag/whatisxp.htm

"Extreme Programming is a discipline of software development based on values of simplicity, communication, feedback, and courage. It works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are and to tune the practices to their unique situation.

In Extreme Programming, every contributor to the project is an integral part of the "Whole Team". The team forms around a business representative called "the Customer", who sits with the team and works with them daily.

Core Practices: Whole Team

Extreme Programming teams use a simple form of planning and tracking to decide what should be done next and to predict when the project will be done. Focused on business value, the team produces the software in a series of small fully-integrated releases that pass all the tests the Customer has defined.

Core Practices: Planning Game, Small Releases, Customer Tests

Extreme Programmers work together in pairs and as a group, with simple design and obsessively tested code, improving the design continually to keep it always just right for the current needs.

Core Practices: Simple Design, Pair Programming, Test-Driven Development, Design Improvement

The Extreme Programming team keeps the system integrated and running all the time. The programmers write all production code in pairs, and all work together all the time. They code in a consistent style so that everyone can understand and improve all the code as needed.

Core Practices: Continuous Integration, Collective Code Ownership, Coding Standards

The Extreme Programming team shares a common and simple picture of what the system looks like. Everyone works at a pace that can be sustained indefinitely.

Core Practices: Metaphor, Sustainable Pace “

XP Distilled, is a publication by Roy Miller and Christopher Collins. In this article the authors discuss the values practices and principles of the Extreme Programming methodology. They discuss how many software projects are late and over budget and how the light weight XP methodology can be implemented to overcome these problems.
The four core values: communication, simplicity, feedback and courage and the 12 practices: the planning game, pair programming, testing, refractoring, simple design, collective code ownership, continuous code ownership, on-site customer, small releases, 40-hour week, coding standards and system metaphor are explained in detail.

[24] Miller and Collins (2001). *XP Distilled,* Retrieved July 15th from http://www.ibm.com/developerworks/java/library/j-xp/?ca=drs-

"Object-oriented programming using the Java language has become immensely popular. It has revolutionized software development to some degree, but recent studies show that half of software development projects are late, and one-third are over budget. The problem isn't the technology; it's the way we develop software. So-called "lightweight" or "agile" approaches, coupled with the power and flexibility of object-oriented languages like the Java language, offer an intriguing solution. The most popular agile approach is called Extreme Programming, or XP, but many people don't really know what it is. Using XP on your Java projects can increase your chances of success dramatically.

Nobody plans to fail. The irony is that approaches created to minimize failure are failing. Misunderstanding software is the root of the problem. Fear really is just a symptom. Existing methodologies were created by smart people who meant well but who forgot the "soft" in software. They assumed that making software is like building a bridge. So they borrowed best practices from various engineering disciplines that apply very well to "hard" things, like bridges. The result is unresponsive development practices based on a Big Design Up-Front (BDUF) mentality, and brittle software that people can't use.

To be blunt, XP (or any other agile methodology) doesn't matter at all. The results it can produce are what count. If an agile approach like XP can help you deliver better software faster and without as much pain, then it is worth considering.

Remember those daunting numbers we mentioned at the start of this article? We believe that using XP to develop object-oriented software has the best chance of changing them for the better. Our experience so far has confirmed that belief."

Extreme Programming Explained: Embrace Change, is a publication by Kent Beck. This book presents a detailed overview and explanation of the Extreme Programming methodology.

It should help the reader discover how to:

- Involve the whole team—XP style
- Increase technical collaboration through pair programming and continuous integration
- Reduce defects through developer testing
- Align business and technical decisions through weekly and quarterly planning
- Improve teamwork by setting up an informative, shared workspace

Some knowledge of other methodologies may help the reader to better understand the differences outlined in this publication.

[25] Kent Beck (2004). *Extreme Programming Explained: Embrace Change,* Second Edition. Addison Wesley Professional.

“Extreme Programming, familiarly known as XP, is a discipline of the business of software development that focuses the whole team on common, reachable goals. Using the values and principles of XP, teams apply appropriate XP practices in their own context. XP practices are chosen for their encouragement of human creativity and their acceptance of human frailty. XP teams produce quality software at a sustainable pace.

One of the goals of XP is to bring accountability and transparency to software development, to run software development like any other business activity. Another goal is to achieve outstanding results—more effective and efficient development with far fewer defects than is currently expected. Finally, XP aims to achieve these goals by celebrating and serving the human needs of everyone touched by software development—sponsors, managers, testers, users, and programmers.

The XP series exists to explore the myriad variations in applying XP. While XP began as a methodology addressing small teams working on internal projects, teams worldwide have used XP for shrink-wrap, embedded, and large-scale projects as well. The books in the series describe how XP applies in these and other situations, addressing both technical and social concerns.

Change has come to software development. However, change can be seen as an opportunity, not a threat. With a plan for change, teams can harness this opportunity to their benefit. XP is one such plan for change."

MySQL: A Lot More Going for It Than No Price Tag is an article written by Steve Renaker where he discusses some of MySQL's notable features, analyzes how it stacks up against other open-source database systems, and demonstrate how to get started using it. Some knowledge of relational database management systems is expected of the reader.

Steve Renaker (n.d.). *MySQL: A Lot More Going for It Than No Price Tag,* Retrieved July 21st from http://archive.devx.com/java/free/articles/Renaker02/Renaker02-1.asp

“In recent years, MySQL has become a very popular open-source alternative to traditional relational database management systems. Companies and developers who at one point might have laid out thousands of dollars for a Sybase or Oracle installation are instead going with MySQL for their database needs. The fact that it's freely available naturally contributes to its popularity, but MySQL has a lot more going for it than just the absence of a price tag—it's fast, lightweight, and very robust.

Database developers who are familiar with other SQL database systems won't find anything shockingly different about MySQL. The MySQL monitor looks very similar to the bare-bones interface of Oracle's SQL*Plus, and most of the ANSI-standard SQL commands will work. MySQL is available for platforms ranging from Windows to Solaris, but the new releases generally appear first for Linux with other ports following at varying intervals.”

Chapter 3 – Methodology

Extreme Programming Methodology (XP)

Overview & Terminology Mapping

Extreme Programming is a lightweight Methodology originated by Kent Beck for rapidly developing high quality software, in the quickest turnaround possible.
It provides the highest value to the customer in terms of quality and implementation time.

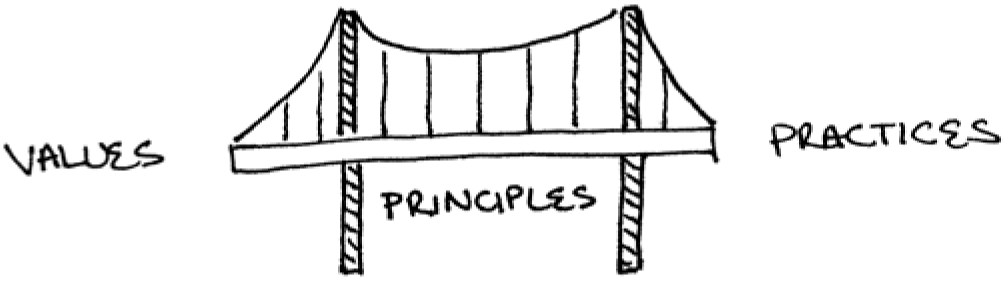


Figure 8 Extreme Programming Overview (Kent Beck, 2004)

According to Kent Beck (2004) XP currently has the following five values Communication, Simplicity, Feedback, Courage and Respect.
Communication – requirements need to be communicated from the customer to the developers and onto the testers. In XP communication is accomplished by collaboration of users and programmers, frequent verbal communication and a lot of feedback. Communication is of the great importance to the success of a project. If each team member does not understand what he/she has to do more formalized methods of communication may be necessary.

Simplicity – XP encourages using the simplest solution and then re-factoring to better designs. The methodology promotes designing for what is necessary at the moment rather than what might be needed next week or next year (J. Donovan Wells. n.d.).

Feedback – relates closely to simplicity and communication. Feedback from the system, customer and development team help to form an overall view of how the system is coming along. Feedback from the system can help establish its quality through unit test. Feedback from the customer can help identify that the system meets there requirements. Feedback from the development team can help estimate the time needed to complete the system or implement change requests (Kent Beck, 2004).

Courage – is needed on the part of the development team to persist with the problem they are trying to solve. They may need to throw out obsolete source code no matter how long it took to write. Developers need to have the courage to raise issues that they find so that the rest of the team is aware of them (Kent Beck 2004).

Respect – members of the XP development team need to have respect for each other, this means not checking in code which will break compilation or cause unit tests to fail. Developers must respect their work by always striving to build the highest quality code (Kent Beck, 2004).

The Principles of XP are based on the values discussed above. They are intended to be employed during the decision making that is required during system development. The important principles are Feedback, Simplicity, Incremental Change and Embracing Change (Kent Beck, 2004).

Feedback – should be sought and given rapidly during all stages of the project, the time between change and feedback should be minimized as much as possible (Ronald E Jeffries, 2001).

Simplicity – all tasks should be viewed with the principle that they are simple to complete. Code for the present and not for the future, keeping things as simple as possible (J. Donovan Wells, n.d.).

Incremental Change – making large changes to software does not always work out with the highest quality results. It is better to have small incremental release, where the customer can see that the changes being made are moving in the right direction. Regression testing is also made easier by the incremental change principle (Kent Beck, 2004).

Embracing Change – it is not always possible to achieve one hundred percent perfect requirements. Once an incorrect requirement is found change is inevitable. Developers must embrace change rather then resist it. It is important that the change is fully understood and any refactoring necessary should be implemented (Kent Beck, 2004).

Planning Game - You have to wait until you have the user stories (lightweight use cases) before you begin production coding. The short term value is that the programmers are relieved of the burden of making decisions that they are unprepared to make. Longer term, the programmer’s only implement stuff the customers are sure they need, and the customers can change the direction of development on a dime. The Planning Game enables most of the other practices by reducing the bulk of what needs to be considered by the programmers at any one time to their customer's immediate needs (Ronald E Jeffries, 2001).

Functional testing - Development cannot be continued until the functional test scores are acceptable to the customer. The short term value is that the programmers know when they are done with the functionality, and the customers are confident that the system works. Longer term, the functional tests prevent regressions and communicate important historical information from a customer perspective. The functional tests back up the unit tests to ensure quality and improve the unit testing process (J. Donovan Wells. n.d.).

Unit testing – Software can't be released until the unit tests are 100% passed. The short term value is that you program faster over hour and greater time scales, the code is higher quality, and there is much less stress. Over the longer term, the unit tests catch integration errors and regression errors, and they communicate the intent of the design independent of implementation details. The unit tests enable refactoring, they drive the simple design, they make collective code ownership safe, and act as a conversation piece enhance pair programming (J. Donovan Wells. n.d.).

Refactoring - Duplicate or uncommunicative code cannot be left around. The short term value is that you program faster, and you feel better because you're understanding and the system as seldom far out of sync. The long term value is that reusable components emerge from this process, further speeding development. Refactoring makes good the bet of Simple Design (Ronald E Jeffries, 2001).

Practices, following are some of the practices which should be employed when developing with the extreme programming methodology.

Simple Design - The best design for the application is the simplest design which will pass all the test cases, has the least amount of code and fewest restrictions and constraints. The code should not be considered complete until it is in this state. In the long run the simplest design will ensure there is less to communicate to other programmers, less code to test and less code to refactor at a later date (J. Donovan Wells, n.d.).

Metaphor - The system should be built around one or a small portion of cooperating metaphors. The benefit of this is that everyone should be confident that they understand what needs to be done and also the metaphors will become a force that tends to tie the design together and make it easier for new team members to come onto the project (Ronald E Jeffries, 2001).

Collective Code Ownership - If a developer comes across some code that could be improved, then the programmer has to stop and make that improvement. The benefits of this are that the whole system begins to improve over time, each developer gets a better understanding of the overall system as they work in areas they are unfamiliar with and refactoring is made easier as team members have a better understanding how different parts of the system integrate (J. Donovan Wells. n.d.).

Coding Standards - Everybody on the team should choose class names and variable names in the same pattern. A tool to format code should be used so that it is always exactly the same style, the team members will begin to automatically use this style.

The benefits of this are easier maintenance, refactoring and easier to bring in a new team member as the code will all have the same style (Ronald E Jeffries, 2001).

Continuous Integration - new code and code changes are integrated with the existing code after a few hours (Ronald E Jeffries, 2001). The code must pass all test cases including regression tests and all conflicts must be resolve before the code is checked in with the existing code. If the above can not be accomplished the code should be discarded. The benefit of continuous integration means that merge conflicts can be resolved earlier, problems are found earlier and it is easier to test large changes as the whole system is compiled frequently (Kent Beck, 2004).

On-site Customer - The team should remain in constant contact with the customer, not just gather the requirements and then ask the customer six months later if that is what they wanted. Constant contact with the customer will help resolve ambiguities, set scope, set priorities, and provide test cases. The benefit of this is confidence that the system is what the customer wants, can be realized earlier in the development and faster turn around of requirement based questions/problems (Ronald E Jeffries, 2001).

Open Workspace - A large room with plenty of places for developers to chat about the system is ideal. Tables and desk with super fast pc's in the middle can be a great help to encourage pair programming and get developers to try out solutions together. In this type of environment it is easier for a developer to call across to another developer and ask for help rather then go solo for hours to solve the problem on their own. The team benefits from this type of communication as an individual's best skills can be called into play when necessary.

Forty Hour Week - it is important for the developers to rest and go home at five in the evening. The XP methodology is an intense method of designing software. Developers need to be at their best to think of issues and problems that may occur. Overtime should be kept to a minimum, XP based projects need a certain amount of stability in the development team. Developers are more likely to stay if long hours are not necessary. Also a well rested programmer is more likely to find valuable refactorings and produce cleaner simpler code (Rational, 1999).

Pair Programming - may seem slower from the out set, but over time the effects of pair programming will be seen in the quality and simplicity of the system. Team moral will be higher as they get to know each other more. The teams should move about three to four times a day. This will ensure that problems will be spread around and will be communicated throughout the team (Ronald E Jeffries (2001)).

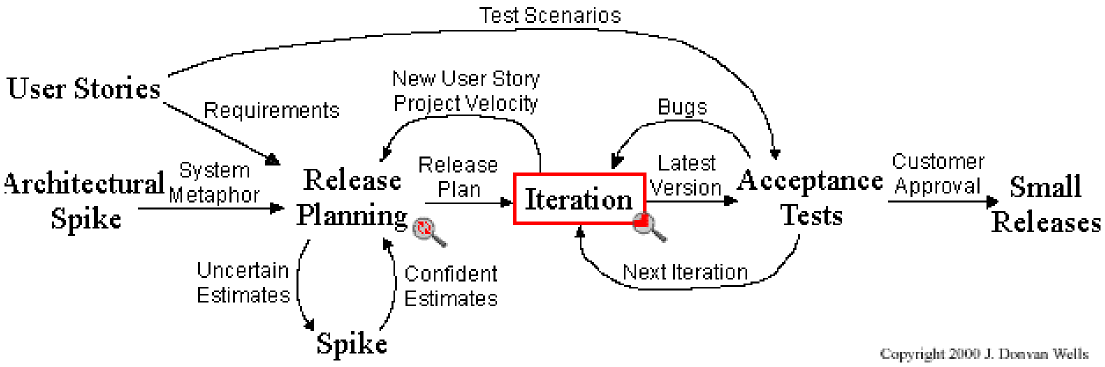


Figure 9 Extreme Programming Test Scenarios

User Stories

According to J. Donovan Wells. (n.d.) user stories are similar to use cases but are not the same. The user stories are used for planning estimates and also instead of a large traditional requirements document.

Candidate Assessment System User Stories

Create New System User

Log onto the system, click on a link to create a new user. Enter the following user details: username, password, name, address, email and phone numbers. The username should be unique within the system. The password should be at least eight characters long.

Create Questions and Answers

Log into the system, click on a link to create a new subject. Then give the subject a unique name within the system. Click on a link to add questions to a subject. The question can have multiple answers or a single answer, the number of answers for the question should be optional. The question should have a selectable difficulty level of easy, medium or hard.

Create Candidate

Log into the system, click on a link to create a new candidate. The following details will be entered into the system. Username, password, email address, the exam duration in minutes, the number of questions in the exam and a selection of subjects that the candidate will be tested on.

Examination

Log into the system, candidate is given a link to begin the exam. The remaining time in the exam should be displayed counting down in seconds. The candidate can iterate forward and backward through the questions, answering questions by selecting either single or multiple answers.

The candidate can change the answer to any given question. Once the time allocated for the exam has expired, the exam will end. A link to end the exam early will complete the exam, not allowing the candidate to change or view answers and will then show the candidate their results.

View Candidate Results

Log into the system, a link will bring the user to a selection of candidates. On selection of a candidate the results of the candidate's previous exams will be available.

Logout

A link will be available to logout of the system. Once the link is selected it will not be possible to access the system again until a username and password is entered to gain access.

Spike Solutions

Spike solutions are used to figure out answers to tough technical and/or design problems that may be realized from close examination of the user stories (J. Donovan Wells. n.d.). Basically a spike solution is a simple program written to explore potential solutions to the problem.

For the candidate assessment system a spike a solution was created to explore using a technology called ICEFaces to provide for the push of data from the server. In other words the server would push information to the client, which is an internet browser (ICEFaces, 2007). This was used to explore the best way to make the exam end, once the allocated time allotted has expired.

Release Planning

At this stage a release planning meeting was held, where the developers, customers and managers reviewed the user stories and estimates, and then agree on a schedule for the release of the system (J. Donovan Wells, n.d.).

Iteration Planning

An iteration planning meeting was called at the start of each development iteration. Each iteration took between one and three weeks. At the planning meeting the customer selected user stories from the release plan in order of the most valuable to the customer first (Ronald E Jeffries, 2001).

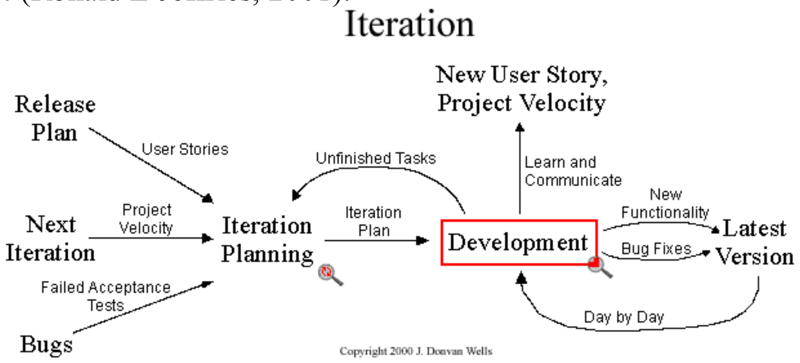


Figure 10 Extreme Programming Iteration

Acceptance Tests

According to J. Donovan Wells. (n.d.) the acceptance tests are created from the user stories. During each iteration the user stories selected during the iteration planning meeting are translated into the acceptance tests.

The user story is not considered complete until it has passed all acceptance tests.

The customer is responsible for the correctness of the acceptance tests and verifying which tests have the highest priority.

The acceptance tests for the candidate assessment system are automated tests written using a technology called JUnit. This makes it possible to run the tests after each new build, any failures in the tests are fixed before the next iteration.

Once all the acceptance tests have been passed the system can be considered to meet all the customer requirements.

Small Releases

After each iteration a small release of the system was generated. This allowed the customer to ensure that the system was heading in the right direction. It also ensured that any integration problems were solved early in the development of the system. (Ronald E Jeffries, 2001).

Diagrams Generated During Development

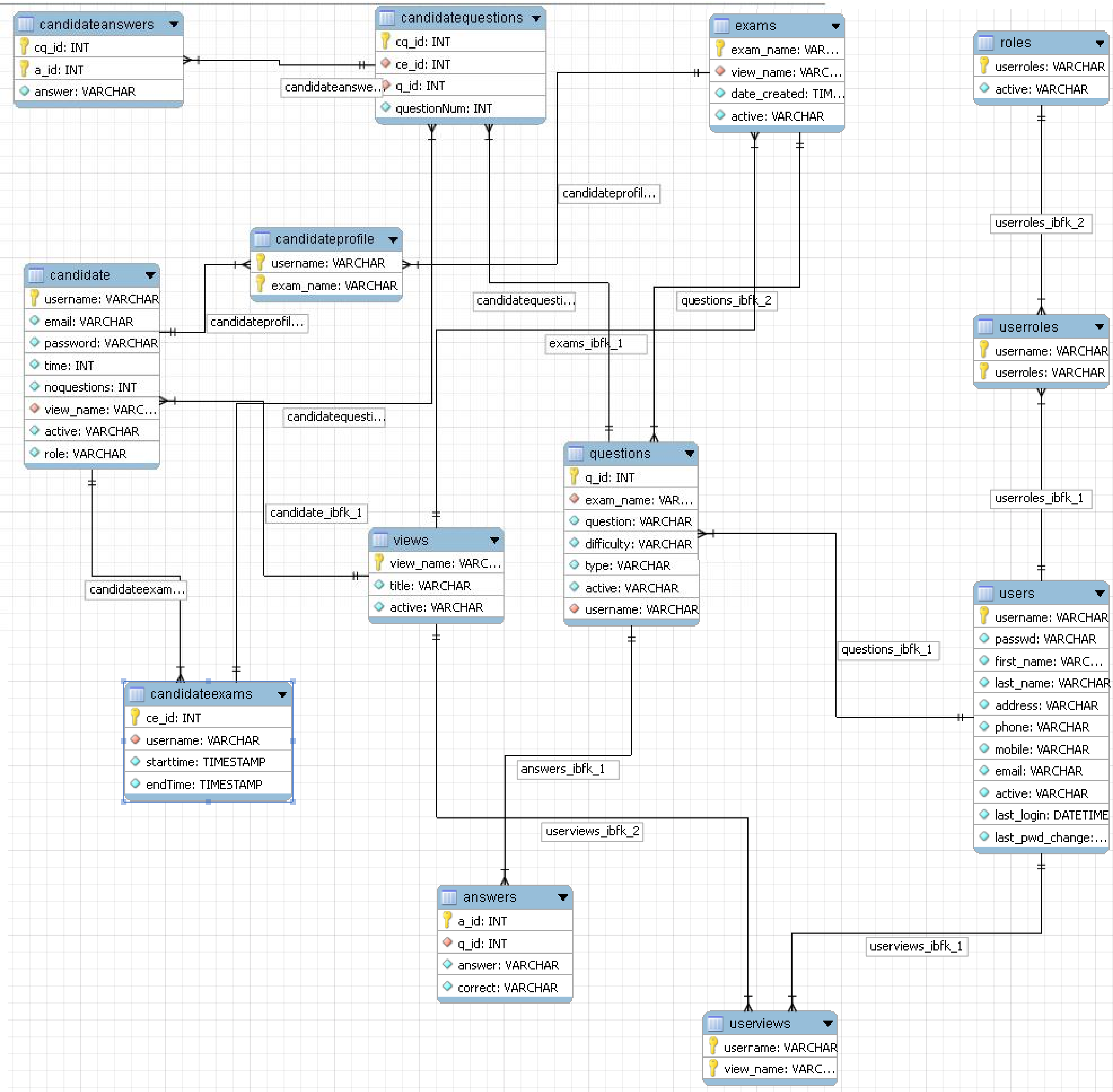


Figure 11 Entity Relationship Diagram

Deployment

Once the final build was complete and all the unit and user acceptance tests had been passed the system was ready for deployment.

The application is deployed in a standardized WAR file which will be built using the well proven ANT build tool. According to Apache (2006) “Ant is different. Instead of a model where it is extended with shell-based commands, Ant is extended using Java classes. Instead of writing shell commands, the configuration files are XML-based, calling out a target tree where various tasks get executed. Each task is run by an object that implements a particular Task interface.” ANT’s flexibility and easy integration with tools such as JUnit made it invaluable during deployment.

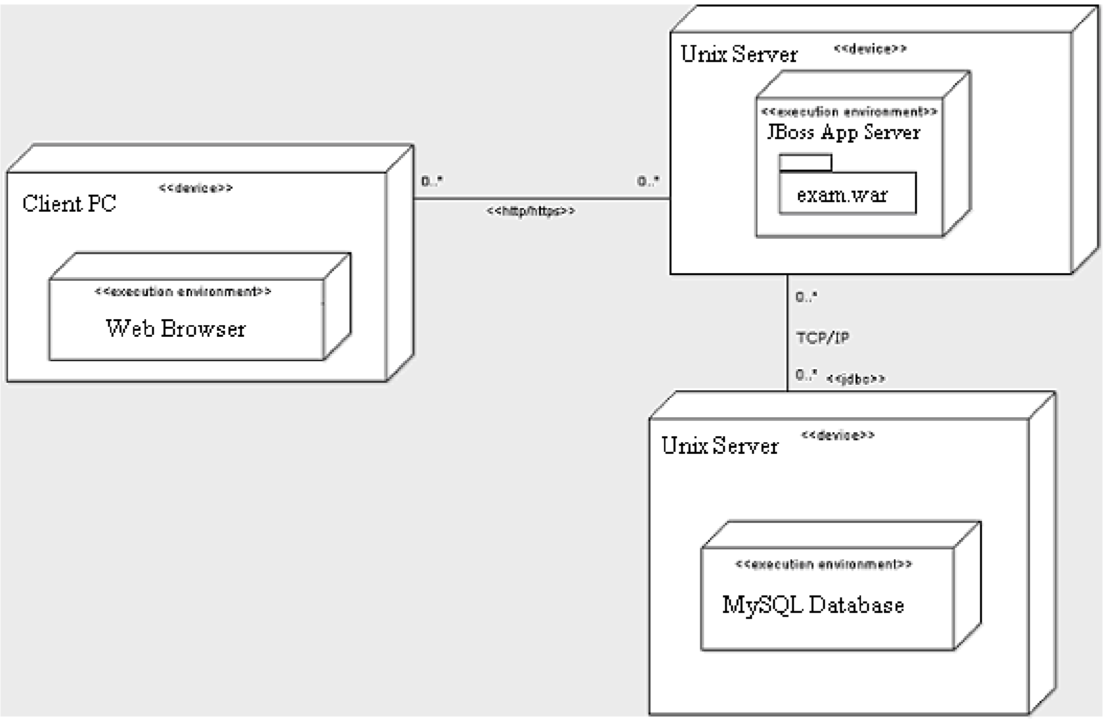


Figure 12 Deployment Diagram

Although the system is completely portable, it was decided to deploy the system on a UNIX based host due to its robustness and the developer’s familiarity with the operating system. According to Juergen Haas (2007) "Unix is extremely flexible and can be installed on many different types of machines, including main-frame computers, supercomputers and micro-computers.

- Unix is more stable and does not go down as often as Windows does, therefore requires less administration and maintenance.
- Unix has greater built-in security and permissions features than Windows.
- Unix possesses much greater processing power than Windows.
- Unix is the leader in serving the Web. About 90% of the Internet relies on Unix operating systems running Apache, the world's most widely used Web server."

A hosting company called eApps was selected to provide the hosting infrastructure. eApps is a high value provider of managed hosting and related services for businesses and organizations throughout the world. eApps is one of the few companies that offers a state of the art hosting platform for commercial grade web sites and applications along with highly personalized, responsive service.

The deployment technology employed is as follows:
JBoss 4 Application Server, this is where the WAR file will be deployed to and dynamic content of the application will be delivered through this container. Jim Rapoza (2004) found that many companies choose the JBoss application server because of its extreme ease of deployment (you basically just unpack it and run it) and because it runs on almost every operating system platform out there. ISVs like it because of its strong support for standards, and its modular design makes it ideal for embedding within other applications.

Apache Web Server, this server will handle the dynamic content such as static html and images.

MySQL 5.0, this is the Database server used to provide persistent storage.

Java 5, is the underlying java virtual machine which is used to execute the application.

IP Address: 69.89.5.121 / Domain name: Devplex.ie

Administration URL: http://www.devplex.ie/exam/admin/admin.iface

Administration Test username/password: super/sup3r

Exam URL: http://www.devplex.ie/exam/pubexam/exam.iface

Exam Test username/password: noelk/noelk

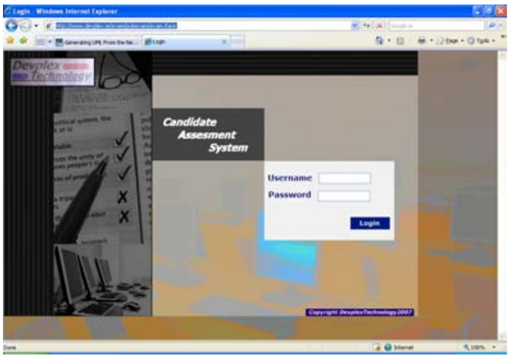


Figure 13 Login page

Schedule

The project schedule was determined by which user stories the customer wanted implemented next. Figure 14 shows a timeline of how long it took to complete each task. The order in which the tasks were completed was dictated by the order of the user stories selected by the customer for completion during each iteration.

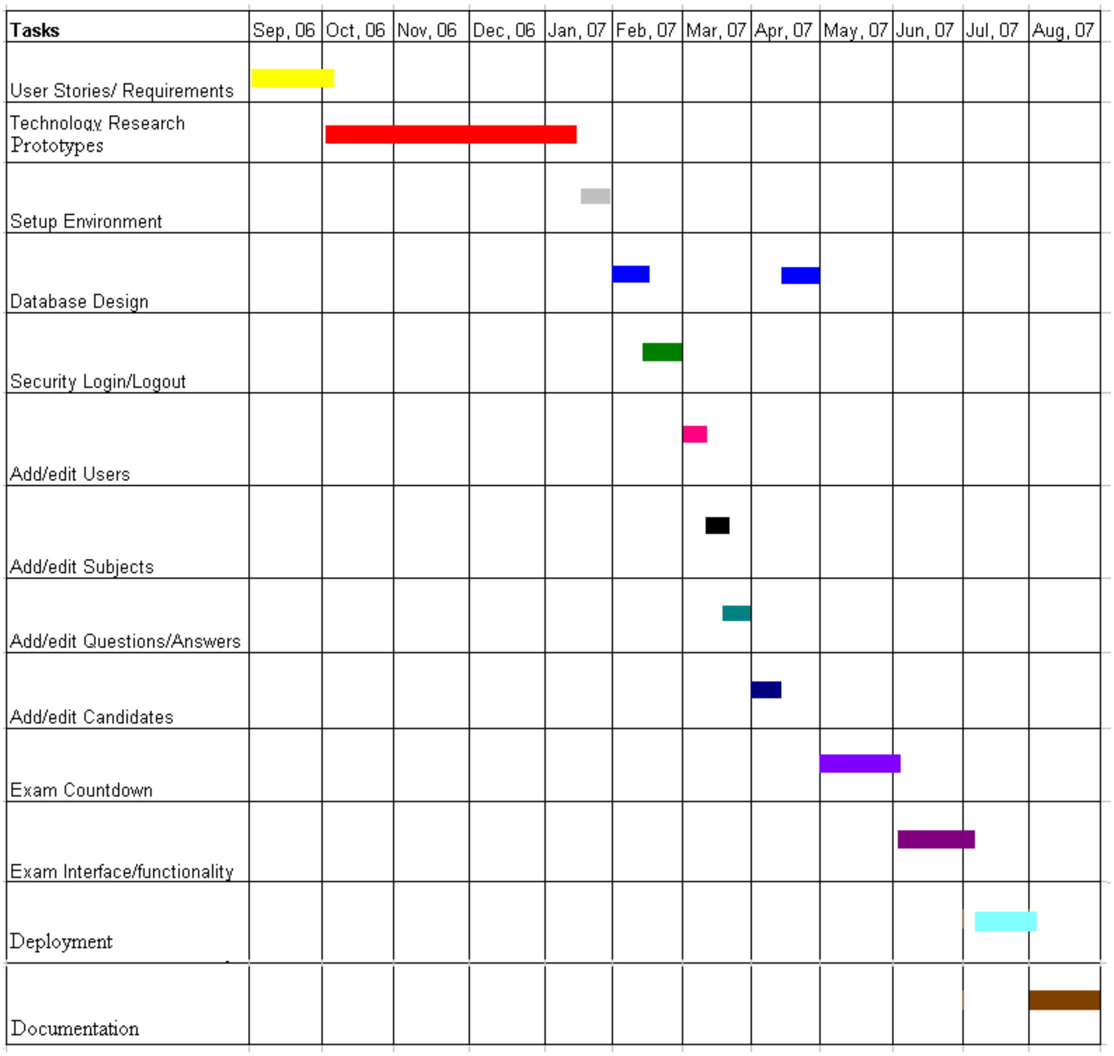


Figure 14 Schedule and Tasks

Chapter 5 – Architecture

Architecture

The underlying architecture of the system comprises of four tiers as depicted in Figure 15.

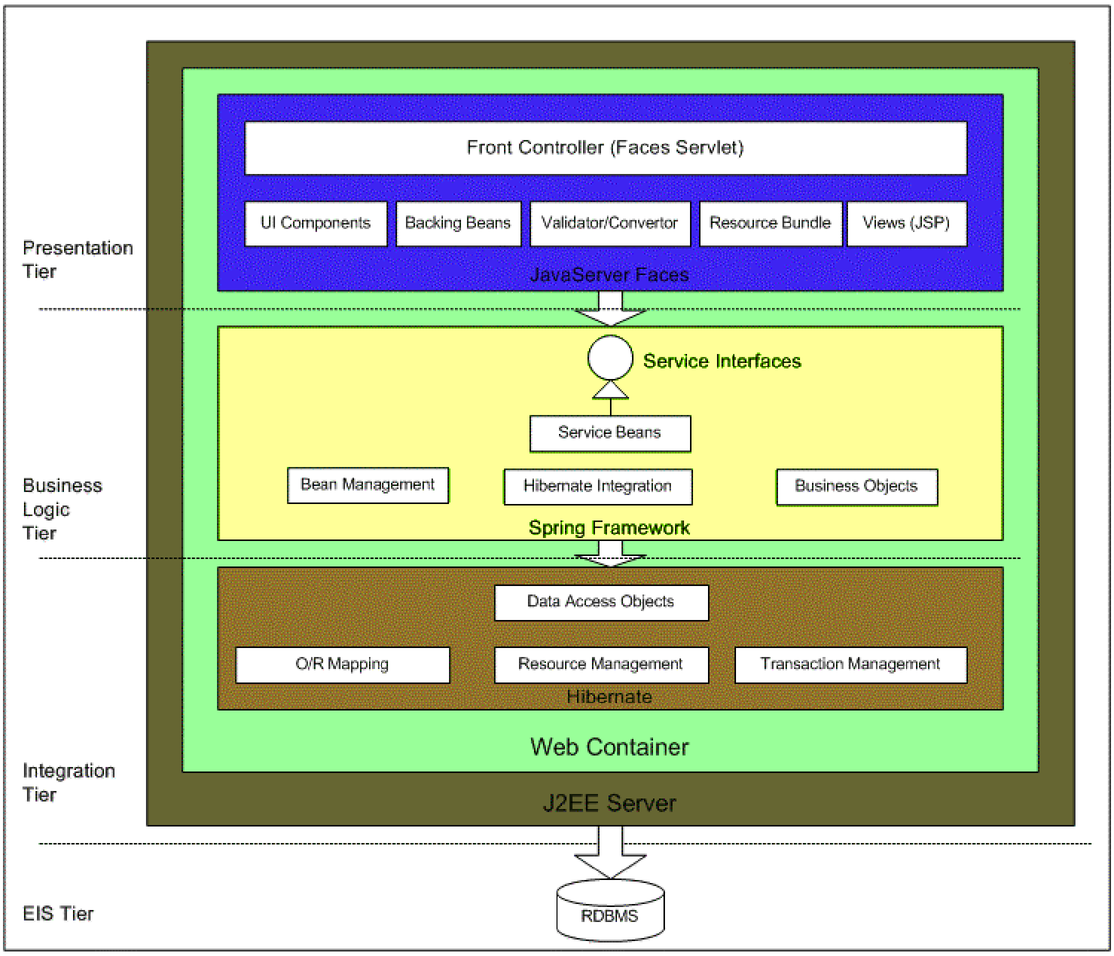


Figure 15 Architectural Tiers (Derek Yang Shen, 2004)

The Java programming language is used as the underlying core technology, the language brings a host of features together that are needed to make this project work. According to David Reilly (n.d.) "the properties that make Java so attractive are present in other programming languages. Many languages are ideally suited for certain types of applications, even more so than Java. But Java brings all these properties together, in one language." some of the properties necessary which are provided by Java are as follows: object-oriented, portable, multi-threaded, automatic garbage collection, secure, network & "Internet" aware, simplicity and ease-of-use.

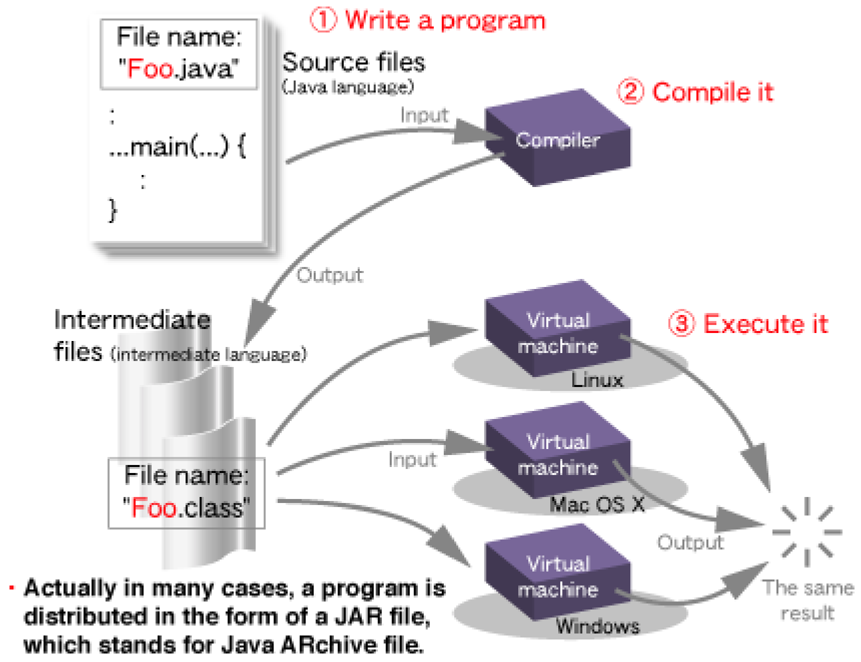


Figure 16 Java Portability (Media Art Online, 2007)

Figure 16 shows how Java creates programs that have to be just written once and executed on almost any environment. First the raw source code is compiled into Intermediate files (bytecode or class files). These intermediate files can then be executed by any JVM with the same results.

An intermediate language lies between a programming language which is understood by human beings and a machine langauge which is executed by a computer. An intermediate language is interpreted by some certain software, generally called an interpreter.

A virtual machine is a software that pretends to be a computer. (Media Art Online, 2007).

This attribute of Java makes CAS portable to almost any operating system or hardware environment.

Presentation Tier

The presentation tier is provided by the JavaServer Faces/ICEFaces framework. This tier contains the technology to present the user interface to the user. It is comprised of the Front Controller, UI Components, Backing Beans, Validator/Convertor, Resource Bundle and Views (Derek Yang Shen, 2004).

The ICEFaces framework works in conjunction with the JSF framework. According to ICEFaces, 2007 "ICEfaces applications development is essentially JSF development, which promotes a component-based architecture using familiar tag-based declarative UI definition and dynamic data binding into the server-resident application data model. Utilizing the ICEfaces Ajax-enabled component suite, which provides all the standard JSF components as well as a complete set of extended components, developers can build a standard JSF application that adheres to the standard server-centric JSF application lifecycle, but benefits automatically from rich Ajax-based features of ICEfaces."

The ICEFaces architecture is depicted in Figure 17.

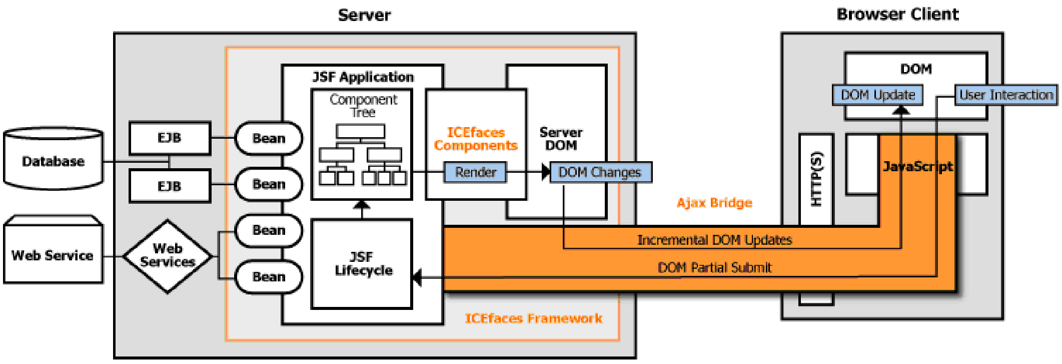


Figure 17 ICEFaces Architcture.

ICEFaces uses AJAX as a method to update the UI components with as little interruption to the user as possible. This method also improves user response times. On web pages without AJAX, the browser requests an entire page from the server. Then, the user clicks on a link or submits a form, at which point the browser sends a new request to the server. The server then sends another page. The Ajax methodology moves away from this page-based model. When the user interacts with a page (clicking a link, submitting a form, and so on), the server sends back a discrete piece of information. Rather than serving up an entire page, the currently loaded page is updated (Jemery Keith, 2007). Figure 18 shows this methodology.

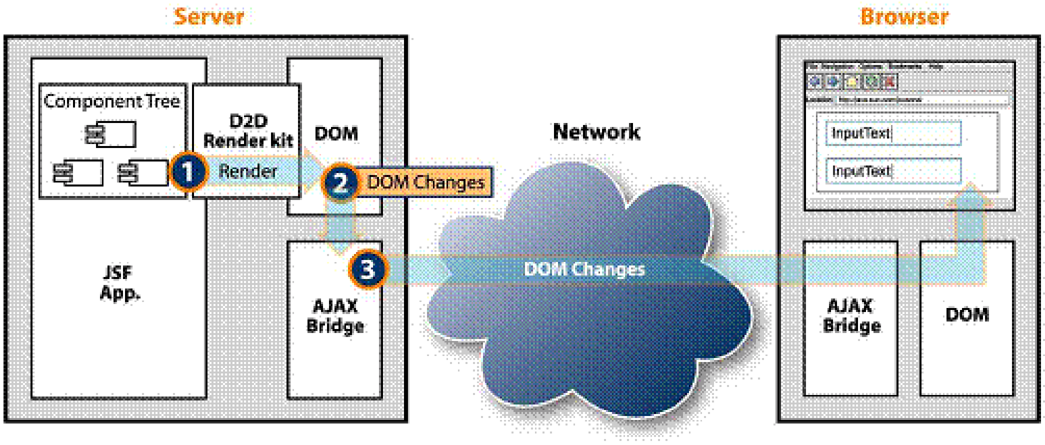


Figure 18 The Direct-To-DOM Model. Using this scheme only portions of the page requiring DOM changes are updated. (Jemery Keith, 2007).

The Front Controller or faces servlet is very similar to the well known Struts Action servlet. JSF uses the MVC design pattern to decouple the view from the model. All requests first go through the faces servlet. The faces servlet handles requests centrally it makes changes to the model and navigates the user to the appropriate view (Crawford and Kaplan, 2003).

JSF uses UI Components as a core part of its architecture. These components enable a high amount of reusability with JSF (Qusay H. Mahmoud, 2004) making the components created for the candidate assessment system reusable in other projects. Custom table components are used in the Candidate Assessment System which give standard html tables the ability to be sorted, scrollable, resizable and the ability to select a row from the table as an actionable event (ICEFaces 2007).

Backing Beans are the core of the JSF framework. They provide a mechanism to communicate with the views. They are tied directly to components; once a component updates in the view this change is reflected directly in the backing bean. This makes managing the state of the application easier than with some other web frameworks (Derek Yang Shen, 2004).

Validator/Convertors are used extensively throughout the candidate assessment system. JSF comes bundled with many inbuilt validators such as validators for numeric only fields or required fields. A custom validator was created to validate email addresses. The convertors are used to convert string values to other types such as integers or floats (Qusay H. Mahmoud, 2004).

The Resource Bundle is used to provide a mechanism to allow the easy maintenance of user messages. The messages can be placed in a resource file and picked up in the application using special tags. The resource file can be updated without any need for compilation or repackaging (Derek Yang Shen, 2004).

The Views in the candidate assessment system are provided by Java Server Pages technology and some static HTML pages. The components are logically laid out in the JSP pages and tied to the backing beans using special syntax. This makes the presentation layer very clean as the JSP pages are quite small and contain mostly HTML making them easy for Graphic Designers to work with.

Business Logic Tier

The business logic tier consists of the business logic and business services that are necessary for the Candidate Assessment System. The business logic tier receives requests from the presentation tier and manages access and requests for data from the Enterprise Information Tier (Derek Yang Shen, 2004).

The underlying framework used in the Business Logic Tier is the Spring Framework. It provides a robust easy to manage method of handling the business services, a clean way to wire the whole application together and an easy method of configuring the application (Rod Johnson, 2007).

Figure 19 depicts the Spring architecture. Only a subset of the Spring frameworks components are used in CAS. The Core component is a fundamental part of the framework and provides the Inversion of Control and Dependency Injection features (Springframework, 2007). It is used in CAS to decouple the configuration and specification of dependencies from the actual programming logic.

The ORM (Object Relational Mapping) Component provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis (Springframework, 2007).

In CAS the ORM package is used to provide the configuration and integration between the application and the object relational mapping tool called Hibernate.

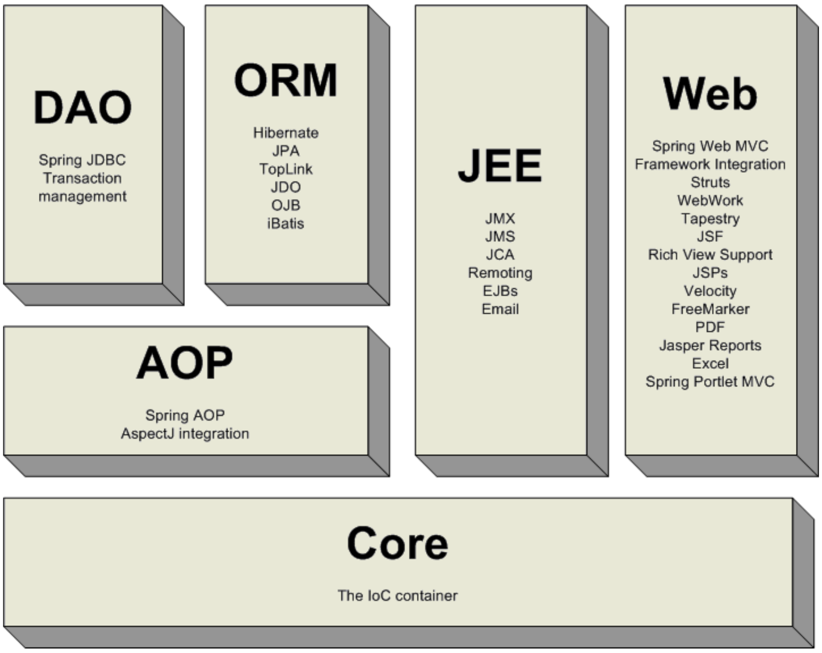


Figure 19 Spring Architecture. (Springframework, 2007)

There are three business services created for CAS, the Admin Service, Exam Service and the User Service. These services interact with the business objects to provide higher level business logic.

Hibernate is used to bridge the gap between the object and relational worlds. It provides and easy mechanism to persist and retrieve the data used around CAS. Spring integrates easily with Hibernate making the handling of Sessions relatively simple (Coulon and Brousseau, 2004).

Integration Tier

Hibernate is the main component behind the Integration Tier. Object relational mapping files are used to integrate the RDBMS with POJO's. Hibernate offers facilities for database connection pooling, data update and retrieval, transaction management, declarative entity relationship management and programmatic and declarative queries (Coulon and Brousseau, 2004).

These features are particularly useful when one wants to avoid using many of the container specific services such as connection pooling and transaction management. Thus making the application easier to port to a different server if the decision was taken at a later stage (Shine Technologies, n.d.).

Figure 20 provides a simple illustration of the Hibernate Architecture. In CAS the XML mapping files are generated by reverse engineering the database using a command line tool provided by Hibernate. The Hibernate properties are configured by the spring framework eliminating the need for the Hibernate.properties file. All persistent storage activates are performed through Hibernate.

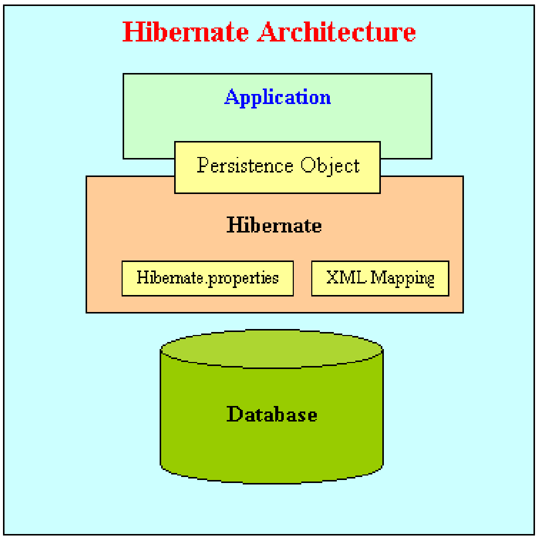


Figure 20 Hibernate Architecture (JavaBeat, 2007).

RDBMS

The MySQL Database provides the persistent storage for CAS. MySQL is a robust and extremely versatile database which provides all the features necessary for CAS. According to Steve Renaker (n.d.) “Database developers who are familiar with other SQL database systems won't find anything shockingly different about MySQL. The MySQL monitor looks very similar to the bare-bones interface of Oracle's SQL*Plus, and most of the ANSI-standard SQL commands will work. MySQL is available for platforms ranging from Windows to Solaris.”

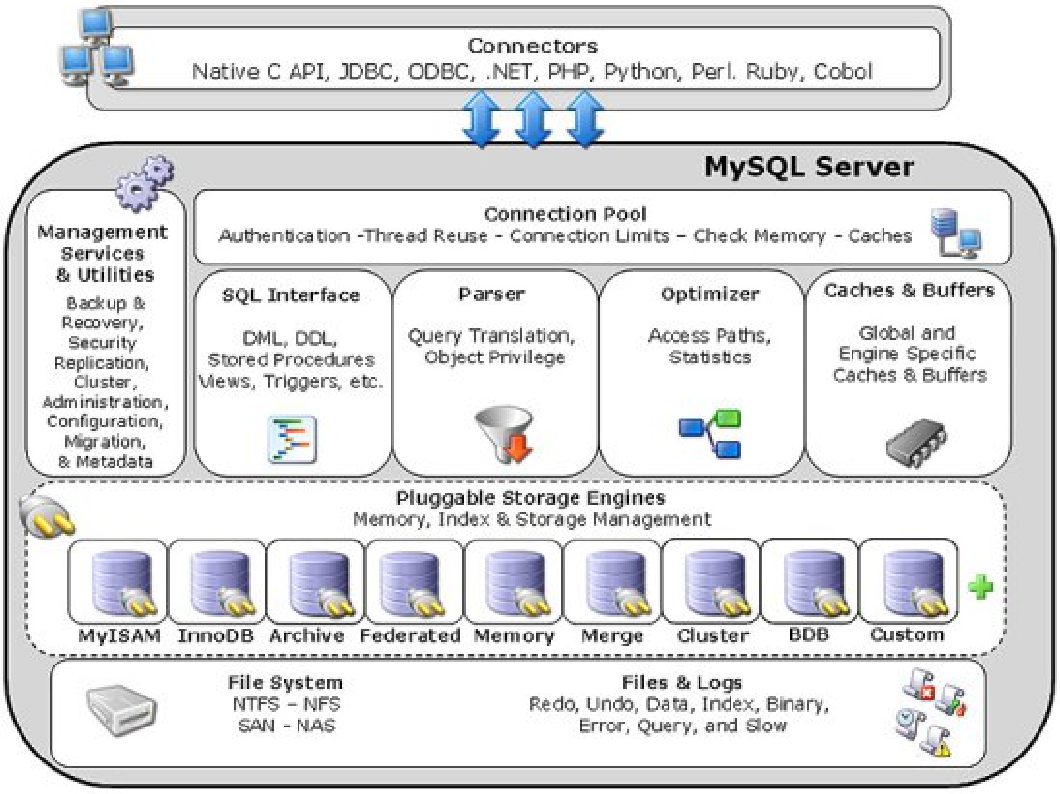

Connectors
Native C API, JDBC, ODBC, .NET, PHP, Python, Perl. Ruby, Cobol
MySQL Server
Management Services & Utilities
Backup & Recovery, Security Replication, Cluster, Administration, Configuration, Migration, & Metadata
Connection Pool
Authentication -Thread Reuse - Connection Limits - Check Memory - Caches
SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.
Parser
Query Translation, Object Privilege
Optimizer
Access Paths, Statistics
Caches & Buffers
Global and Engine Specific Caches & Buffers
Pluggable Storage Engines
Memory, Index & Storage Management
MyISAM
InnoDB
Archive
Federated
Memory
Merge
Cluster
BDB
Custom
File System
NTFS - NFS
SAN - NAS
Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query, and Slow

Figure 21 MySQL Architecture Pluggable Storage Engine Architecture is both flexible and modular (Robin Schumacher, 2004).

Figure 21 shows that MySQL is a full featured database this will ensure the CAS will be able to scale and perform as demands are increased and technologies evolve without the need to change database vendors. It also shows that using MySQL has provided a certain amount of future proofing for CAS.

Security

Because CAS is to be an internet based application, security is a big concern. According to Gary McGraw (2004), “Internet-enabled software applications are a commonly (and too easily) exploited target, with software's ever-increasing complexity and extensibility adding further fuel to the fire. By any measure, security holes in software are common, and the problem is growing.” In order to provide strong security a framework called JAAS provides the underlying security framework for CAS. According to Kyle Gabhart, (2002) "The Java Authentication and Authorization Service (JAAS) is a set of packages that enables services to authenticate and enforce access controls upon users. It implements a Java version of the standard Pluggable Authentication Module (PAM) framework, and supports user-based authorization." JAAS simplified the Java security development by putting an abstraction layer between the application and the underlying authentication and authorization mechanism. This independence from algorithms and platform allows the system to use different security mechanisms without modifying the application-level code.

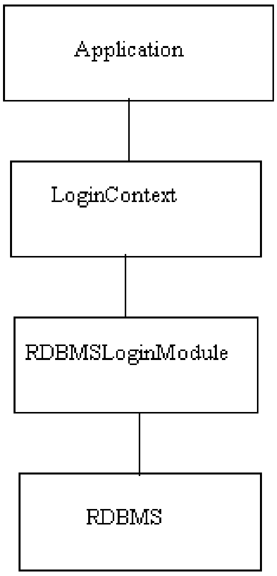


Figure 22 JAAS Overview

The application deals primarily with the login context this provides all the details that the application requires. The login context in turn can interface with one or more login modules (Kyle Gabhart, 2002).

JAAS provides some reference login module implementations. For CAS the RDBMSLoginModule reference implementation is used. It is configured by a simple configuration file, which provides the SQL to lookup the user's password and also what roles the user has.

The following lists the security configuration for CAS. It is configured in web.xml.

```
<login-config>
                <auth-method>FORM</auth-method>
                <form-login-config>
                        <form-login-page>/login.iface</form-login-page>
                        <form-error-page>/errors.html</form-error-page>
                </form-login-config>
</login-config>
<security-role>
     <description>Administrators</description>
     <role-name>Admin</role-name>
 </security-role>
 <security-role>
     <description>Super User</description>
     <role-name>Super</role-name>
 </security-role>
 <security-role>
```

```
    <description>Exam Taker</description>
    <role-name>ExamTaker</role-name>
 </security-role>
<security-constraint>
    <display-name>AdminConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>Admin</web-resource-name>
      <description>Only for administrators</description>
      <url-pattern>/admin/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <description/>
      <role-name>Admin</role-name>
      <role-name>Super</role-name>
    </auth-constraint>
  </security-constraint>

  <security-constraint>
    <display-name>AdminConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>Admin</web-resource-name>
      <description>Only for administrators</description>
      <url-pattern>/exam/*</url-pattern>
```

```
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <description/>
            <role-name>Admin</role-name>
            <role-name>Super</role-name>
        </auth-constraint>
    </security-constraint>

    <security-constraint>
        <display-name>AdminConstraint</display-name>
        <web-resource-collection>
            <web-resource-name>Admin</web-resource-name>
            <description>Only for administrators</description>
            <url-pattern>/user/*</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <description/>
            <role-name>Admin</role-name>
            <role-name>Super</role-name>
        </auth-constraint>
    </security-constraint>
```

```
<security-constraint>
   <display-name>CandidateConstraint</display-name>
   <web-resource-collection>
      <web-resource-name>Admin</web-resource-name>
      <description>Only for administrators</description>
      <url-pattern>/candidate/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
   </web-resource-collection>
   <auth-constraint>
      <description/>
      <role-name>Admin</role-name>
      <role-name>Super</role-name>
   </auth-constraint>
</security-constraint>

<security-constraint>
   <display-name>ResultsConstraint</display-name>
   <web-resource-collection>
      <web-resource-name>Admin</web-resource-name>
      <description>Only for administrators</description>
      <url-pattern>/results/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
```

```
    </web-resource-collection>
    <auth-constraint>
      <description/>
      <role-name>Admin</role-name>
      <role-name>Super</role-name>
    </auth-constraint>
  </security-constraint>

  <security-constraint>
    <display-name>CandidateConstraint</display-name>
    <web-resource-collection>
      <web-resource-name>ExamTaker</web-resource-name>
      <description>For exam takers</description>
      <url-pattern>/pubexam/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <description/>
      <role-name>Admin</role-name>
      <role-name>Super</role-name>
      <role-name>ExamTaker</role-name>
    </auth-constraint>
  </security-constraint>
```

Chapter 6 – The Application

The Administration Subsystem

The administration subsystem is used by administrators and HR personnel to administer the system. The administration system can be used to add/edit questions, users, candidates, exams and view candidate results.

The system uses the ICEFaces Direct to DOM technology, this provides a superior user experience, in that the web pages never refresh as the web page document is updated in place. According to Laurance Moroney (2005) "ICEfaces adds this Direct-to-DOM model to the JSF framework. The result is that only elements that change need to be delivered to the client-side and rendered. Changes to the state of the GUI, then, do not require complete page refreshes. This allows for a number of rich features that aren't typically available with standard Web-based apps"

A special component has been provided on the page to give the user an indication that activity is occurring in the background (ICEFaces, 2007), much the same as how the progression bar works which can be found at the bottom of many internet browsers such as Internet Explorer and FireFox. The component is shown if figure 23.

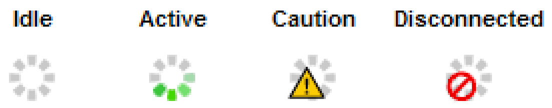
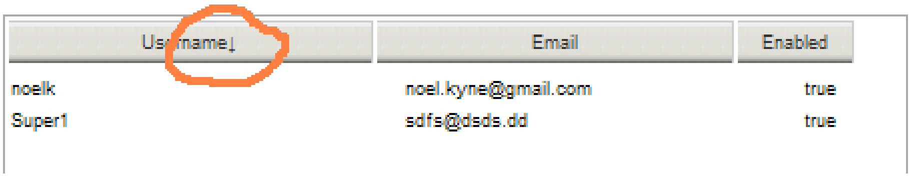


Figure 23 Connection Status (ICEFaces, 2007)

All of the tables in the application feature the client side sort function which allows the contents of the table to be sorted with no round trip to the server thus saving the user time and bandwidth. The user simply clicks on the column heading to sort ascending or descending respectively. The command sort header component is used in conjunction with the data table component. The command sort header renders a clickable column header facet allowing the user to toggle the sort order of data in the table, either ascending or descending based on the values in the column (IceFaces, 2007).



Username↓	Email	Enabled
noelk	noel.kyne@gmail.com	true
Super1	sdfs@dsds.dd	true

Figure 24 Sort Order

Login page – the entry point to the administration system, Users must enter a username and password to access the system.

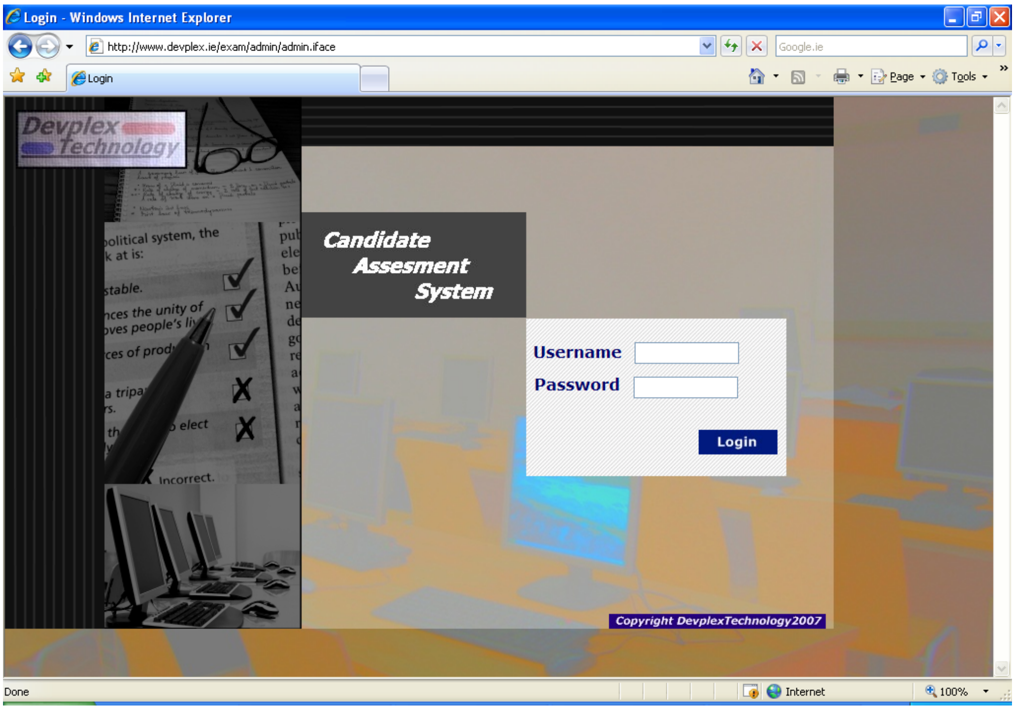
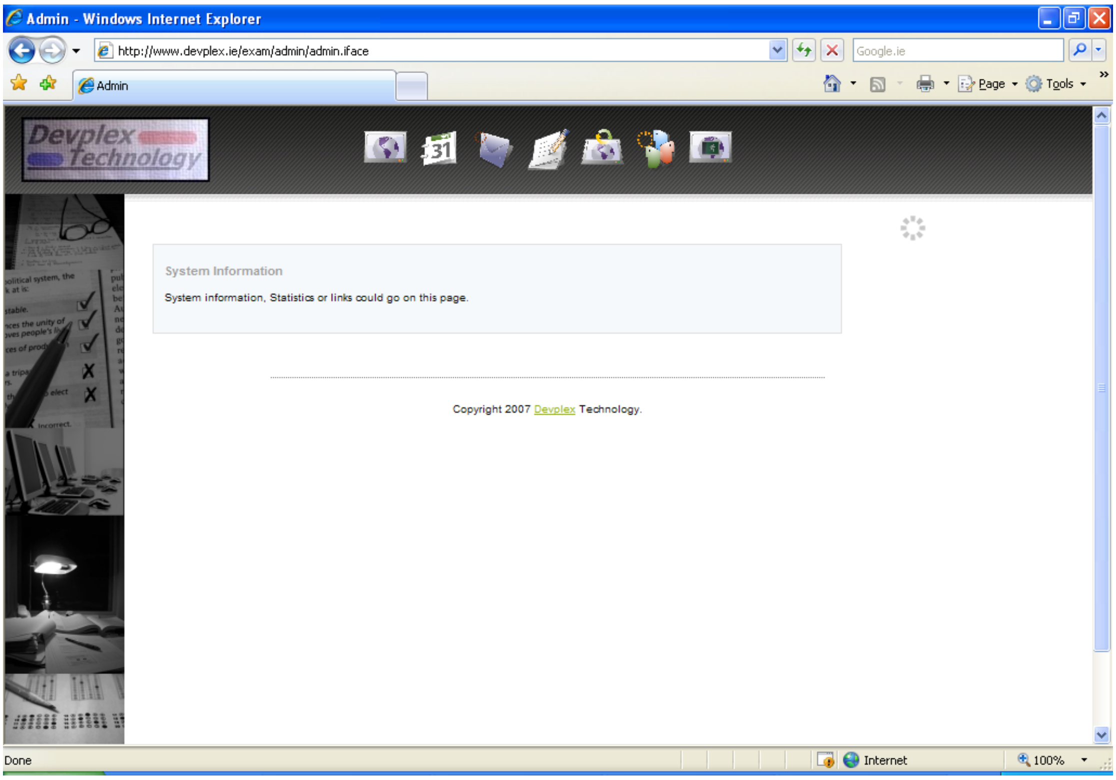

Login - Windows Internet Explorer
http://www.devplex.ie/exam/admin/admin.iface
Google.ie
Login
Page
Tools
Devplex Technology
Candidate Assesment System
Username
Password
Login
Copyright DevplexTechnology2007
Done
Internet
100%

Figure 25 Administration Login Page

Landing page – once logged in successfully the user is presented with the landing page. This page can be used to present system maintenance information or other company information.


Admin - Windows Internet Explorer
http://www.devplex.ie/exam/admin/admin.iface
Google.ie
Admin
Page
Tools
Devplex Technology
31
Done
Internet
100%

System Information

System information, Statistics or links could go on this page.

Copyright 2007 Devplex Technology.

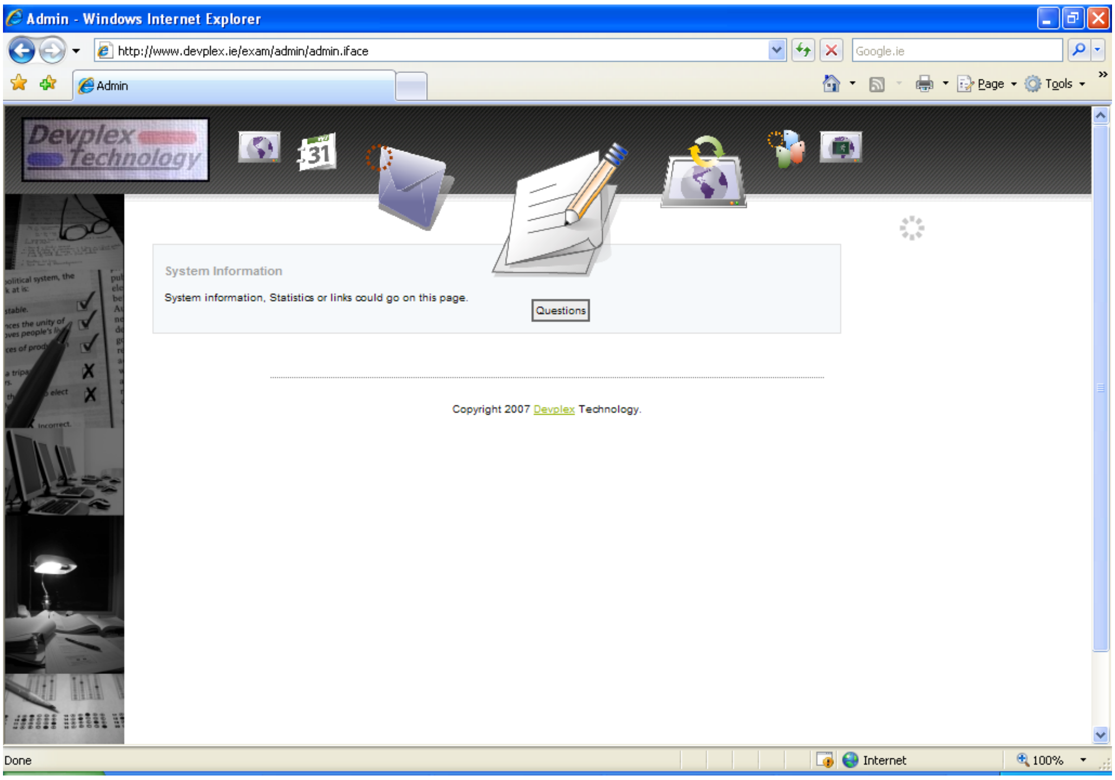
Figure 26 Administration Landing Page

Top Menu Navigation (Dojo toolkit -- Fisheye) is generated using a module from an open source JavaScript toolkit called dojo. The toolkit is designed to ease the rapid development of JavaScript- or Ajax-based applications and web sites. According to Jenifer Twidwell. (2005) “Navigating around a web site or application is like commuting. You have to do it to get where you need to go, but it's dull, it's sometimes infuriating, and the time and energy you spend on it just seems wasted. Can't you do something better with your time, like playing a game or getting some actual work done?

The best kind of commuting—all else being equal—is none at all. Having everything you need right at your fingertips, without having to travel somewhere, is pretty convenient. Likewise, keeping most tools "within reach" on an interface is handy, especially for intermediate-to-expert users (i.e., people who have already learned where everything is). Sometimes you do need to put lesser-used tools in separate dialog boxes, where they don't clutter things up; sometimes you need to group things onto different pages so the interface makes sense. All this is fine, as long as the "distances" that a user has to travel remain short." To overcome this problem in the candidate assesment system the fisheye component is used to provide a nice easy to use menu system for the users all functions are simple one click away.

The menu system can be used to access the following functions:

Users, Questions, Candidates, Exams and Results



Admin - Windows Internet Explorer
http://www.devplex.ie/exam/admin/admin.iface
Google.ie
Admin
Page
Tools
Devplex Technology
31
Questions
Done
Internet
100%

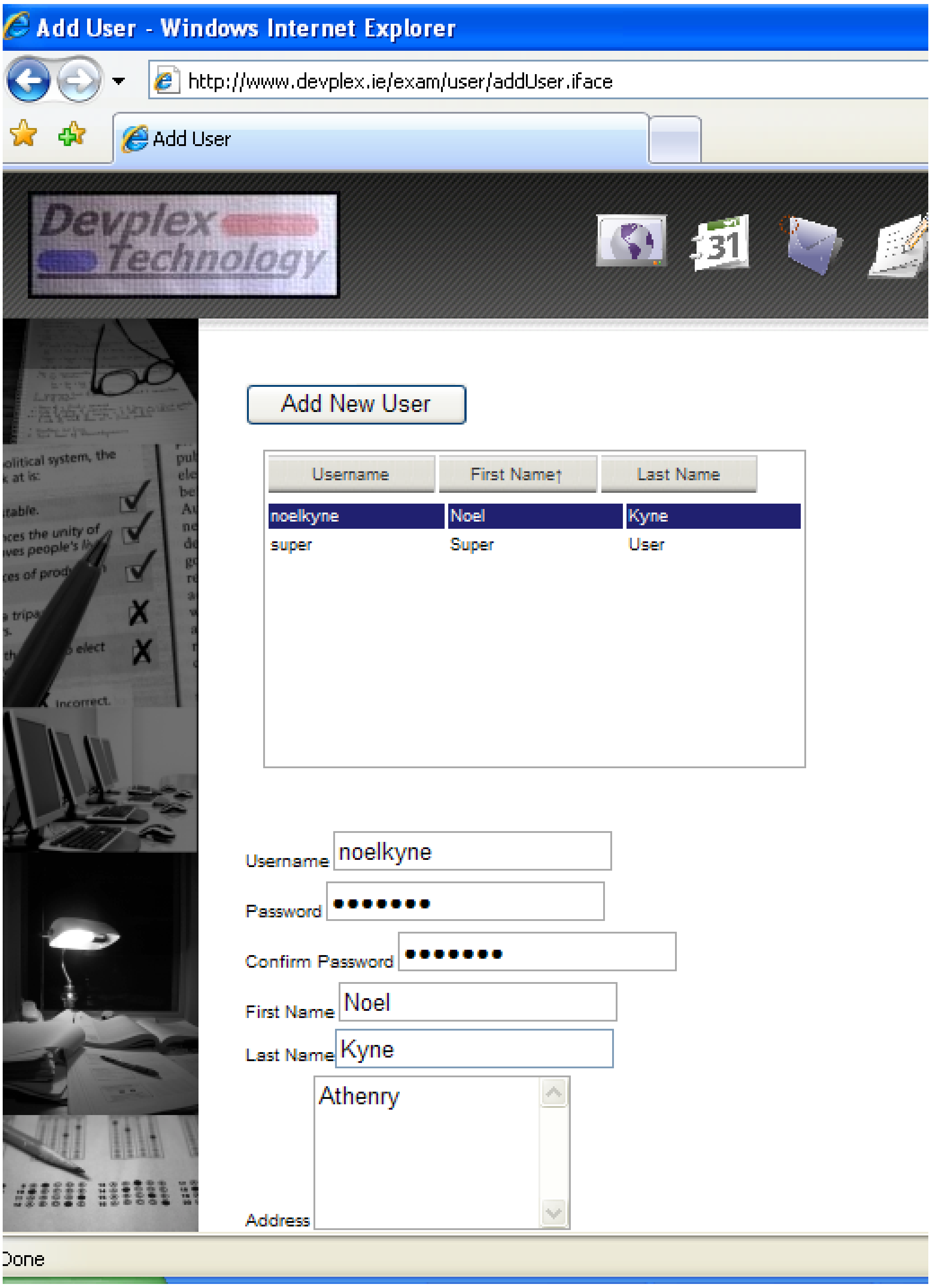
System Information

System information, Statistics or links could go on this page.

Copyright 2007 Devplex Technology.

Figure 27 Fisheye Component

New/Edit user – figure 28 shows how the system allows administrators to create new users and edit existing users, who can perform different roles within the system.



Add User - Windows Internet Explorer
http://www.devplex.ie/exam/user/addUser.iface
Add User
Devplex Technology
31
Add New User
Done

Username	First Name↑	Last Name
noelkyne	Noel	Kyne
super	Super	User

Username noelkyne

Password •••••••

Confirm Password •••••••

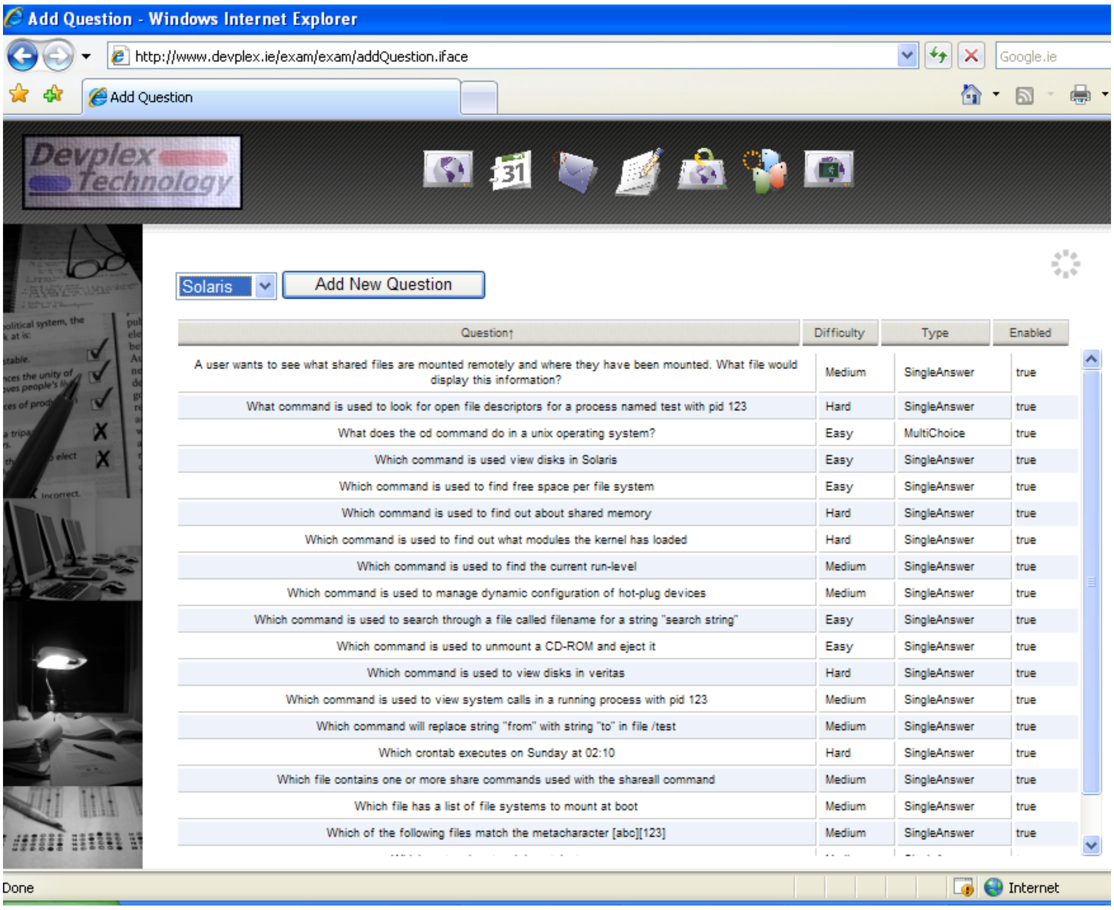
First Name Noel

Last Name Kyne

Address Athenry

Figure 28 Administration Add User

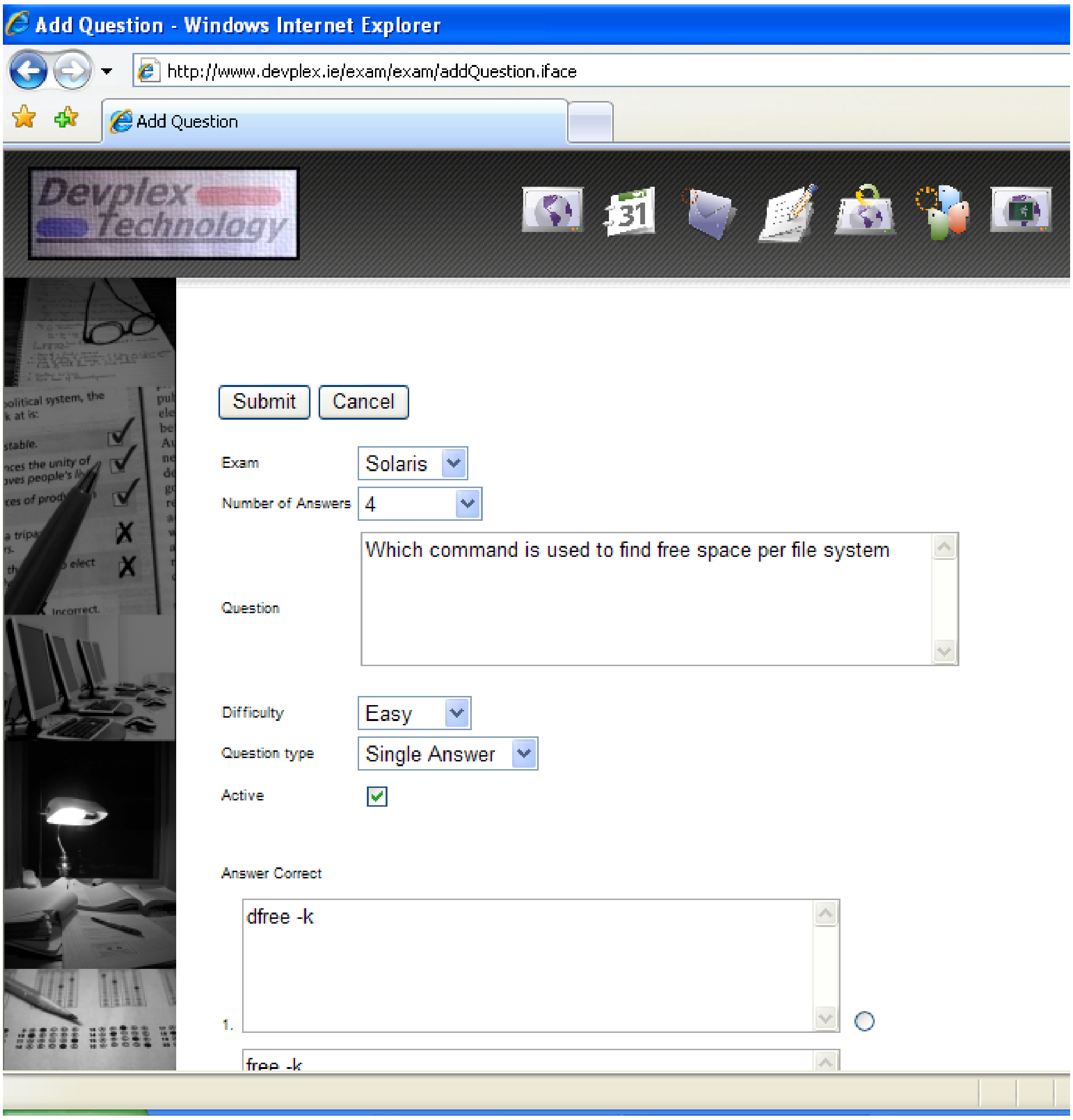
Edit/View Question – this screen shown in figure 29 allows the user to edit questions. It also allows the user to read the questions and see the answers. This is helpful when conducting phone and face to face interviews.


Add Question - Windows Internet Explorer
http://www.devplex.ie/exam/exam/addQuestion.iface
Google.ie
Add Question
Devplex Technology
Solaris
Add New Question
Done
Internet

Question↑	Difficulty	Type	Enabled
A user wants to see what shared files are mounted remotely and where they have been mounted. What file would display this information?	Medium	SingleAnswer	true
What command is used to look for open file descriptors for a process named test with pid 123	Hard	SingleAnswer	true
What does the cd command do in a unix operating system?	Easy	MultiChoice	true
Which command is used view disks in Solaris	Easy	SingleAnswer	true
Which command is used to find free space per file system	Easy	SingleAnswer	true
Which command is used to find out about shared memory	Hard	SingleAnswer	true
Which command is used to find out what modules the kernel has loaded	Hard	SingleAnswer	true
Which command is used to find the current run-level	Medium	SingleAnswer	true
Which command is used to manage dynamic configuration of hot-plug devices	Medium	SingleAnswer	true
Which command is used to search through a file called filename for a string "search string"	Easy	SingleAnswer	true
Which command is used to unmount a CD-ROM and eject it	Easy	SingleAnswer	true
Which command is used to view disks in veritas	Hard	SingleAnswer	true
Which command is used to view system calls in a running process with pid 123	Medium	SingleAnswer	true
Which command will replace string "from" with string "to" in file /test	Medium	SingleAnswer	true
Which crontab executes on Sunday at 02:10	Hard	SingleAnswer	true
Which file contains one or more share commands used with the shareall command	Medium	SingleAnswer	true
Which file has a list of file systems to mount at boot	Medium	SingleAnswer	true
Which of the following files match the metacharacter [abc][123]	Medium	SingleAnswer	true

Figure 29 Administration Edit/View Question

Add New Question – this screen shown in figure 30 is presented when the user is editing a question or adding a new question. It dynamically adds or removes the form components required if the user wishes to change the number of available answers for the question. The user can cancel the operation if required.


Add Question - Windows Internet Explorer
http://www.devplex.ie/exam/exam/addQuestion.iface
Add Question
Devplex Technology
Submit
Cancel

Exam: Solaris

Number of Answers: 4

Question: Which command is used to find free space per file system

Difficulty: Easy

Question type: Single Answer

Active: ☑

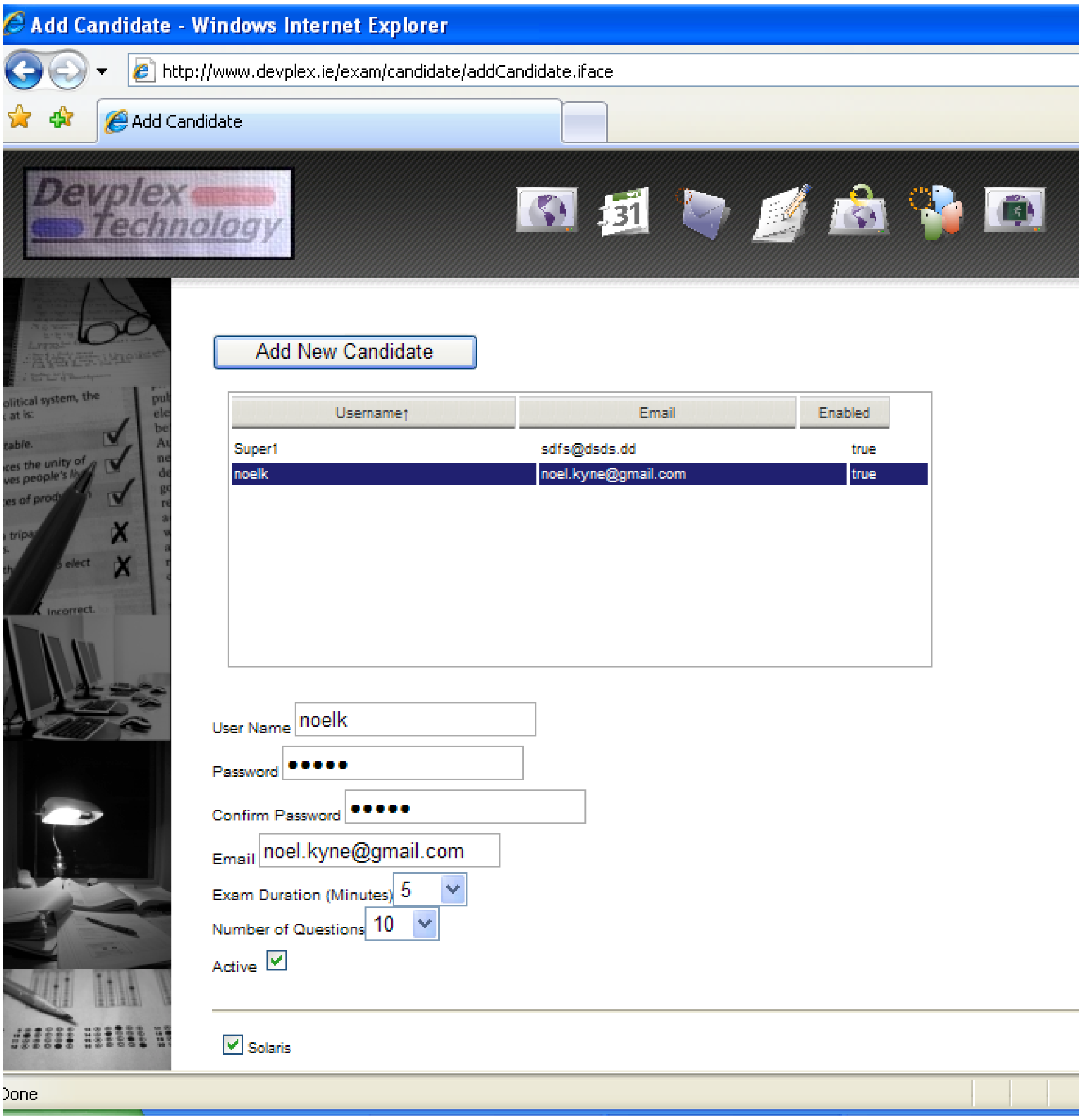
Answer Correct

1. dfree -k ○

free -k

Figure 30 Administration Add Question

Create/Edit Exam Candidate, this screen shown in figure 31 allows the user to edit/add new candidates. These candidates will subsequently be able to log into exam sub system and take the assessment that has been set up for them on this page. The candidate personal details are stored and the exam details are taken at this time such as number of question, time and the categories which the questions will be taken from.


Add Candidate - Windows Internet Explorer
http://www.devplex.ie/exam/candidate/addCandidate.iface
Add Candidate
Devplex Technology
31
Add New Candidate
Done

Username↑	Email	Enabled
Super1	sdfs@dsds.dd	true
noelk	noel.kyne@gmail.com	true

User Name noelk

Password •••••

Confirm Password •••••

Email noel.kyne@gmail.com

Exam Duration (Minutes) 5

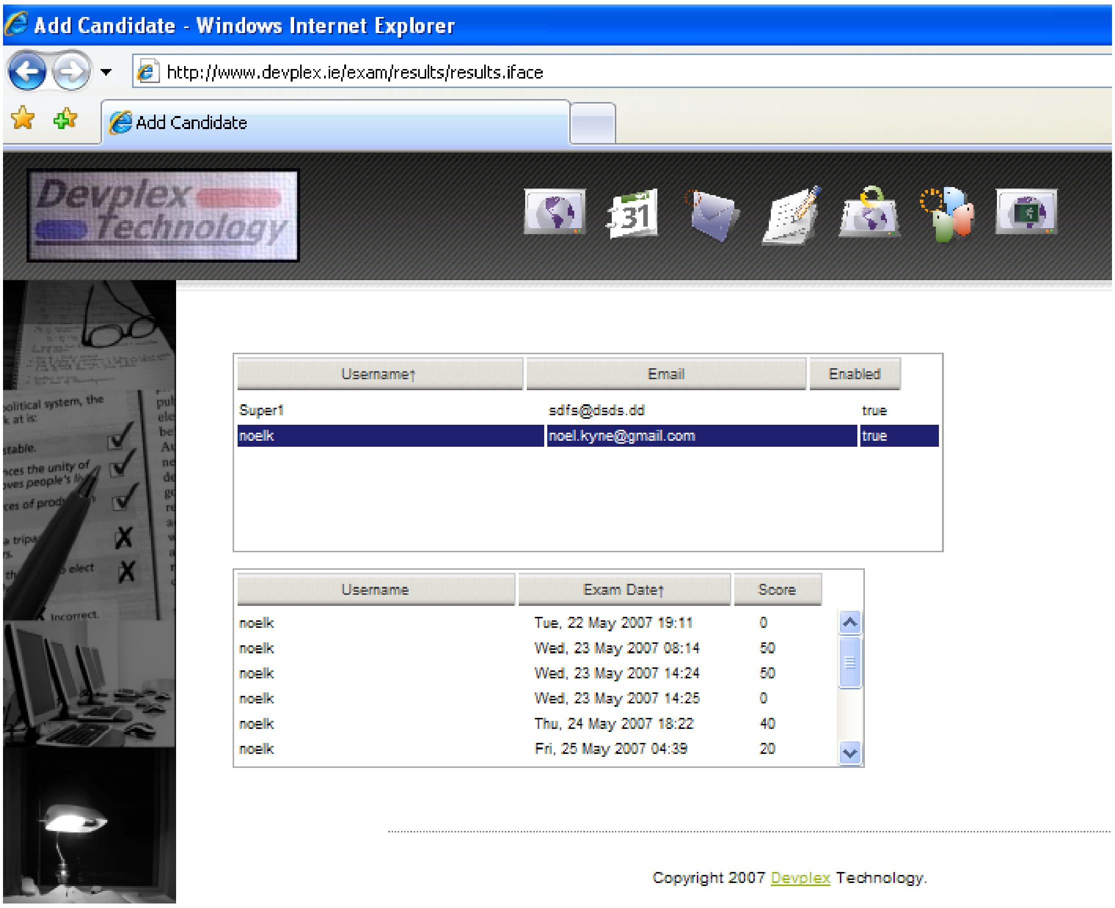
Number of Questions 10

Active ☑

☑ Solaris

Figure 31 Administration Add/Edit Candidate

View Candidate Results – this screen shown in figure 32 allows the user to view the candidate's exam results. The result of each exam taken can be viewed in the second table.


Add Candidate - Windows Internet Explorer
http://www.devplex.ie/exam/results/results.iface
Add Candidate
Devplex Technology

Username↑	Email	Enabled
Super1	sdfs@dsds.dd	true
noelk	noel.kyne@gmail.com	true

Username	Exam Date↑	Score
noelk	Tue, 22 May 2007 19:11	0
noelk	Wed, 23 May 2007 08:14	50
noelk	Wed, 23 May 2007 14:24	50
noelk	Wed, 23 May 2007 14:25	0
noelk	Thu, 24 May 2007 18:22	40
noelk	Fri, 25 May 2007 04:39	20

Copyright 2007 Devplex Technology.

Figure 32 Administration View Candidate Results

The Exam Subsystem

Login page – this screen shown in figure 33 is used by candidates to login and access the exam system. The candidate must enter a valid username and password to gain access.

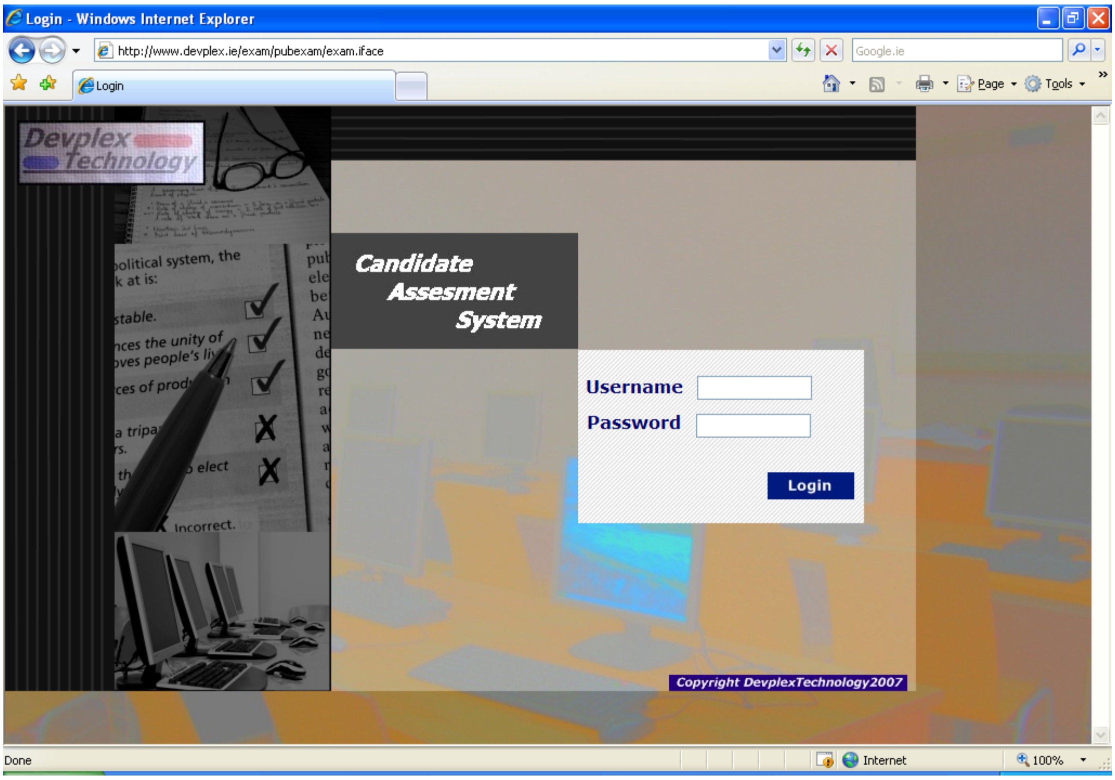

Login - Windows Internet Explorer
http://www.devplex.ie/exam/pubexam/exam.iface
Google.ie
Login
Page
Tools
Devplex Technology
Candidate Assesment System
Username
Password
Login
Copyright DevplexTechnology2007
Done
Internet
100%

Figure 33 Exam Login Page

Landing page – once login is successful, the landing page shown in figure 34 is presented to the candidate. From here the candidate can click on the Start Exam button which will start the time counting down and present the first question to the candidate.

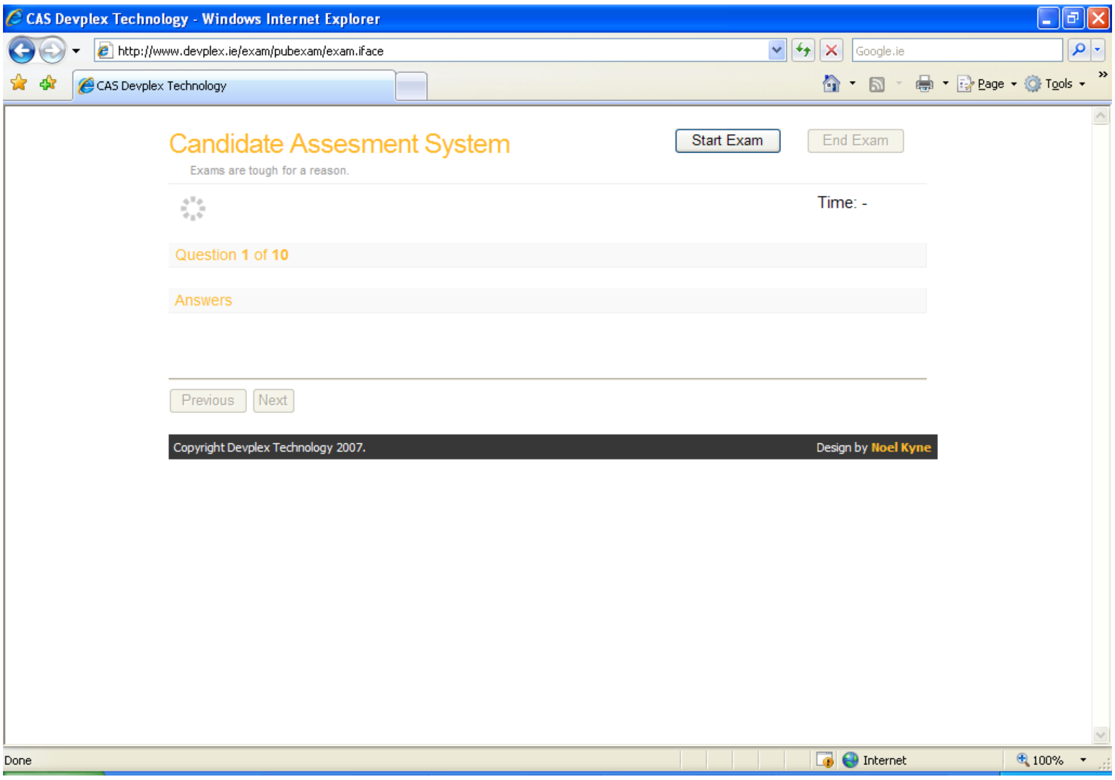
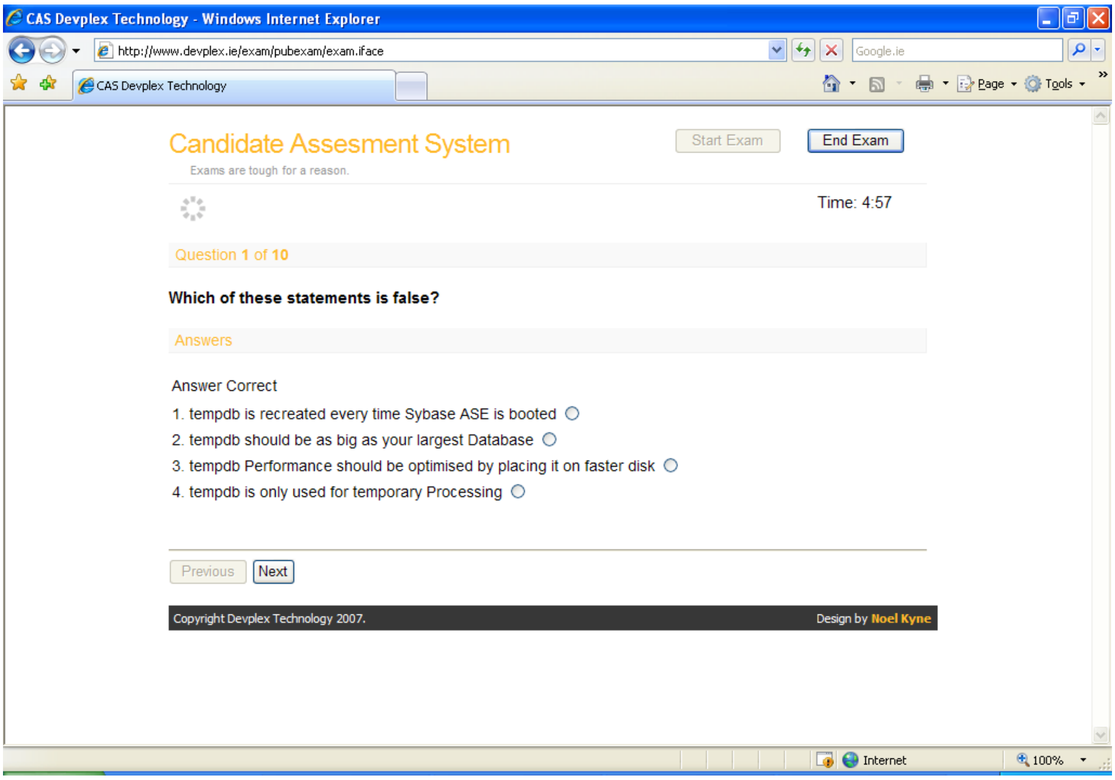

CAS Devplex Technology - Windows Internet Explorer
http://www.devplex.ie/exam/pubexam/exam.iface
Google.ie
CAS Devplex Technology
Page
Tools
Candidate Assesment System
Exams are tough for a reason.
Start Exam
End Exam
Time: -
Question 1 of 10
Answers
Previous
Next
Copyright Devplex Technology 2007.
Design by Noel Kyne
Done
Internet
100%

Figure 34 Exam Landing Page

Exam in progress – this screen shown in figure 35 presents the questions to the candidate. The candidate can end the exam early by clicking the End Exam button. The time is shown counting down on the top right hand side of the screen. The previous and next buttons can be used to iterate forwards and backwards through the questions.



CAS Devplex Technology - Windows Internet Explorer
http://www.devplex.ie/exam/pubexam/exam.iface
Google.ie
CAS Devplex Technology
Page
Tools
Start Exam
End Exam
Previous
Next
Internet
100%

Candidate Assesment System

Exams are tough for a reason.

Time: 4:57

Question 1 of 10

Which of these statements is false?

Answers

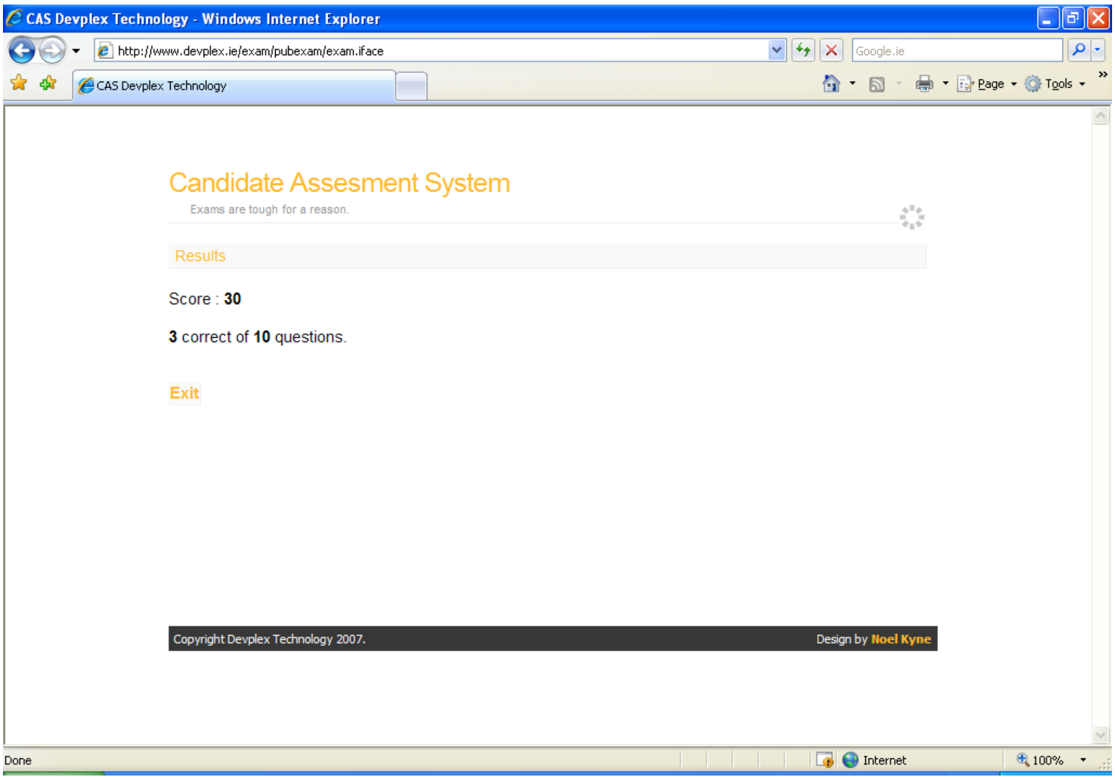
Answer Correct

1. tempdb is recreated every time Sybase ASE is booted ○
2. tempdb should be as big as your largest Database ○
3. tempdb Performance should be optimised by placing it on faster disk ○
4. tempdb is only used for temporary Processing ○

Copyright Devplex Technology 2007. Design by Noel Kyne

Figure 35 Exam Display Page

Exam over – Shown in figure 36 is the screen presented to the candidate once they have clicked the End Exam button or once the exam time has expired. It shows the candidates score and the number of questions they got correct. Clicking on the Exit button brings the user to the login page.



CAS Devplex Technology - Windows Internet Explorer
http://www.devplex.ie/exam/pubexam/exam.iface
Google.ie
CAS Devplex Technology
Page
Tools
Done
Internet
100%

Candidate Assesment System

Exams are tough for a reason.

Results

Score : **30**

3 correct of **10** questions.

Exit

Copyright Devplex Technology 2007. Design by **Noel Kyne**

Figure 36 Exam Results Page

Chapter 7 – Project History

How the Project Began

The project began by the HR manager requesting that all Senior Engineers working at Devplex Technologies submit questions with answers that could be used to test potential candidates in the areas that the submitting engineer was most proficient in. Once the questions were gathered and reviewed, they had to be typed into MS Word documents.

At this point, it was realized that this was a cumbersome method of maintaining the data and also difficult to retrieve and utilize the data.

The first idea was to store the data in a database and develop a desktop application for fast retrieval and maintenance of the data. This would provide an easy method for the HR manager to quickly retrieve questions and ask the candidate the questions.

The first idea evolved into another idea which involved providing a way to automate the method in which the candidates were assessed for their technical ability. This idea evolved into the candidate assessment system which allows candidates to be assessed on site or remotely, with varying levels of difficulty and to assess a candidate across a number of subjects.

Milestones in the Project

There were many milestones in the project. Significant milestones are documented as follows:

Milestone 1 – Selection of technologies

A number of different technologies were investigated and researched to find the best solution for the system. Once technologies that could achieve the systems requirements were found, they were evaluated by producing small prototypes which gave a better indication of there suitability. Once Java, JSF, Hibernate, Spring, JBoss and MySQL were selected it signaled that the project had reached milestone 1.

Milestone 2 – Completion of the Administration Subsystem

This was an important milestone, as it meant completion of the administrator functionality of the system. Once this section was functionally complete it was provided to the customer for user acceptance testing. Once the subsystem was deemed to be functionally and technically complete, milestone 2 had been reached. The completion of milestone 2 made it easier to develop and test the exam subsystem.

Milestone 3 – Completion of the Exam Subsystem

Once the exam subsystem was completed, this meant that the core functionality of the system was completed and the whole system could be tested as a complete functional unit.

Milestone 4 – Setup of the deployment environment

This was another important milestone for the system as it meant that the system would be available over the internet where real testing with a small user group could begin. The development environment was rented from a company called eApps. It was important to make sure that all functionality worked as expected as there were some slight differences between the development, test and production environments.

Milestone 5 – Handover

This was the final project milestone and it signaled the completion of this phase of the project. All user and functional tests were completed and passed before Milestone 5 was reached. A baseline of the application was taken and a final build was produced. The system was then handed over to Devplex Technologies for live production usage.

Changes to the project plan

Changes to the project plan were minimal. There was a delay of one week setting up the hosting and acquiring the domain name, but this did not impact the overall project timeline. Environment problems meant that testing with the small user group had to be scheduled for a week later than was planned. This time was made up again due to the high quality of the system and completing this testing a week earlier then scheduled.

Project Evaluation against Objectives

Once the system was fully completed it was possible to evaluate the project against its objectives. It was found that the system met all of the requirements defined.

The user interface exceeded the expectations of Devplex Technologies.

Performance requirements and stability were all within expected ranges.

Findings/Results

This project revealed a number of different methods with which it could be implemented.

Standalone Desktop Application model – This method of implementation can provide the most rich user interface experience and best overall response time for the user. But the difficulties posed with installing these fat clients on the end users machines ruled this out as the implementation model given that many of the end user may only use the system once. Also the problems associated with upgrading these fat clients further proved it was not the best solution.

Client/Server model – This method of implementation provides the highest level of portability. It also easily maintainable and scaleable as code for this type of implementation is deployed at a single location. The application is accessed over the internet through a standards based web browser which is now available on virtually all devices. The main problem with this model is that the user interface can be difficult to equal the same standard as standalone applications. Traditionally another problem with the client server model is response time. In many cases user only had dial up internet connections which was too slow for delivering large amounts of data, this affected performance greatly.

The client/server model was chosen for CAS as it provided the best all around solution given the requirements. The performance problems are largely mitigated by the wide availability of broadband and higher speed internet connections that are used today.

The user interface difficulties were solved using a technology used specifically for improving user interface design called JSF. Used in addition to JSF is ICEFaces. This technology further improves the user experience by using AJAX to update components on the page without reloading the entire page. This gives a more standalone application feel to the web application.

Using the Extreme Programming Methodology proved to be of great benefit to the overall development of the system. The methodology provides the constant feedback that the system being developed is what the customer wants. It also ensures that the code being developed is of the highest standard and that no unnecessary work is being produced.

Lessons Learned

Many lessons were learned during the development of CAS. These are documented as follows:

ICEFaces is a new cutting edge technology which is changing with each new release. CAS was started with an early release of ICEFaces. Upgrading to the newer releases during the project proved difficult as the changes in ICEFaces were significant. Also its integration with MyFaces Tomahawk was not fully complete which made it difficult to utilize many of the excellent components created by the MyFaces project.

Learning a new technology can be very time consuming. During the early attempts developing with a new technology, the quality is invariably lower than code developed later in the development cycle. It is important that as developers become more confident with a technology, that they go back and re-factor the earlier implemented portions.

The development environment was completely underpowered for this project and had to be upgraded soon after development commenced. A Pentium 3 based PC with 512MB of RAM is unsuitable for running the MySQL database, JBoss application server and Eclipse IDE all at the same time. This environment was upgraded to a Pentium 4 PC with 1,536MB RAM which was a more suitable environment.

Unit test should always be written in parallel to the methods and classes being developed. It is very difficult to come back later and write unit tests for old classes as valuable test cases will be missed or a unit test may return a false positive as the writer may not fully understand how the class being tested should work. In this case the unit test would just prove that the class works as it currently does which may be erroneous.

Future Plans

This first implementation of CAS can be looked at as phase one. In a second phase of the project the following features should be implemented.

Company Views – this functionality would make the system available to multiple companies. Essentially it would provide a view for each company that wanted to use the system. It would be possible to then create a pool of standard questions that any company could use, while maintaining independence of questions that the company create specifically for their own view. Companies would not be able to view details outside of their own view thus maintaining privacy. This would increase the user base considerably and may require clustering of the application server to handle the increased volume.

CV Scan – This functionality would provide a method of scanning CV's and automatically creating a profile for candidates. The system would be configured to accept email from a particular email address. It would then scan the attached CV and generate a profile for the candidate. Following this the system would send an email to the candidate with their login details. The candidate would then be able to login remotely and take the exam. This functionality would greatly reduce the level of HR interaction and would have the ability to create a great pool of resources to choose from.

Reporting – Reporting functionality could be provided to generate the top 10 candidates for a given profile, the number of questions available and question statistics. The question statistics could provide information such as the number of times a question was answered correctly/incorrectly this could help with determining the difficulty level of a question.

Chapter 8 – Conclusion

Conclusion

Devplex Technology Limited required a system to help them to interview, assess and select new employees for job opportunities within their company. The current interview techniques required many man-hours gauging whether candidates were technically suitable for opportunities, by manually interviewing people. An optimal method would be to design and develop a system which would examine an interviewee on their expertise in certain technologies. This project provided an excellent opportunity to research the technologies available in solving such a unique problem.

J2EE was chosen as the core platform for the project due to its platform neutrality and its range of freely available frameworks and components which can simply plug-in. The Java programming language was an easy choice as no other language can provide the same level of flexibility when used over the internet. Also the language has a large range of freely available components which can be used off-the-shelf.

The JBoss application server was chosen over the many other available servers due to its completeness in terms of J2EE standards and for its ease of deployment. There is also the possibility of clustering the servers to provide greater capacity and robustness if it is needed in the future.

The database chosen is the MySQL version 5.0. This version of MySQL supports foreign key constraints, stored procedures and views which previous version had no support for. While other database vendors such as Oracle or Sybase produce arguably higher performance, it was found that MySQL provided all the functionality needed and performance was adequate given that the database transactions of the application would not produce considerable load. Given the above and no license fee required made MySQL the right choice.

The object relational mapping tool called Hibernate was chosen to bridge the gap between object oriented design in Java and relational database design in MySQL. Hibernate provides an excellent tool to reverse engineer the database. This was especially useful as a proper normalized database structure could be created in the traditional way and a head start could be made on the Hibernate mapping files as the existing schema structure could be reverse engineered by Hibernate to create the mapping files.

JSF is a framework used for building user interfaces for web applications. It was chosen for this project because the user interface is particularly important and needs to be as flexible as possible. For this project, it was decided to use another flavor of JSF called ICEFaces. ICEFaces is based on the existing JSF framework but also adds an AJAX bridge and direct to DOM technology. This technology gives ICEFaces based web pages a far superior user experience. One of the more important reasons that ICEFaces was chosen is its ability to enable server/application initiated content rendering to the browser. This feature was used to provide the exam time duration count down for the candidate’s exam. The seconds counting down are pushed directly from the server using a feature called Ajax push. ICEfaces pioneered Ajax Push allowing developers to build collaborative applications like never before.

The Spring Framework is one of the cornerstones of the application, chosen for its non invasive architecture it provides enterprise application power with simple plain old java objects. The Spring framework provided an easy mechanism for configuration application through java beans. This greatly reduces the over head of loading in configuration files at application startup time.

A build tool called ANT was chosen to build the application source code and also package the application into a WAR file. ANT provides a very simple platform independent build mechanism which is easy to configure and extend. It was also used to execute automated unit tests.

The Extreme Programming Methodology provided a very stable method of planning and tracking the progress of the project. It allowed for verification early on in the development that the system being built is what the customer actually wanted. It also ensures that the code being written is of a high quality and that unit tests are continuously being written and updated; this is an element that can often fall behind in other methodologies as developers come under pressure to deliver working code.

This project consisted of a complex set of very specific requirements. In each area the very best of breed technologies were considered and implemented.

The Candidate Assessment System provides Devplex Technologies Limited with a tool which will effectively and efficiently pre-assess potential candidates for job opportunities within its company. This hugely benefits Devplex Technologies, as it provides them with an excellent indication as to whether a potential employee is knowledgeable and capable in the areas they require. It works to help them select the most suitable candidate for the position.

The Candidate Assessment System will help Devplex Technologies to save the amount of time that technical employees will spend interviewing the unsuitable candidates for the company. The system will also store the results of an examination and the candidate details which may assist HR to find candidates that may be suitable for future positions.

From a personal perspective, the design and development of the Candidate Assessment System has taught this author a tremendous amount. This project has provided an insight into the life cycle of a project from the analysis and requirements gathering stage through to deployment and maintenance. From the technologies that have been used in the development of this project, this authors' knowledge and capability has increased immensely, especially in the technologies ICEFaces/JSF and Hibernate which are new and emerging technologies.

The Extreme Programming Methodology provided this author with a knowledge and confidence in utilizing this methodology and applying it to other projects.

The database and application servers' set-up and configuration presented this author with experience in these technologies. Working with a hosting company, for the first time, and configuring and setting up the database, web server and application server with the hosting company served its own experience. This project also required that a

DNS be configured for the domain which this author had no previous experience of, therefore enabling the acquiring of new skills and knowledge.

The Candidate Assessment System has exceeded all its expectations by providing Devplex Technologies Ltd. with a successful system for gauging the expertise of perspective employees. The System has achieved all project objectives:

- Remote accessibility

  CAS is accessible through www.devplex.ie/exam/admin/admin.iface all that is required is an internet connection and a standard web browser.

- Secure

  JAAS provides an authentication and authorization service for CAS, this service ensures that all users have logged in successfully and that only users with adequate permissions have access to the restricted functions.

- Easy/Intuitive to use

  ICEFaces provides a superior user interface for the user. It combines standard web page controls with AJAX giving the user a progressive feeling as they navigate around the system. The screen never goes blank while new content is being loaded ensuring that the user always feels in control.

- Low maintenance

  No substantial maintenance is required only adding new questions or candidates as necessary.

- Low running costs

  CAS runs on a single server, this reduces the overhead of purchasing/renting several servers. No license fees are required as all the technology is freely available.

While achieving all this, the system has provided a catalyst for researching, learning and developing new and emerging technologies and resources.

References

[1] Gary McGraw. (2006) *Software Security: Building Security in.* Pearson Education, Inc.

[2] Jenifer Twidwell. (2005). *Designing Interfaces*. O'Reilly

[3] Jeremy Keith. (2007). *Bulletproof Ajax.* New Riders

[4] Jim Farley. (2001). *Microsoft .NET vs. J2EE: How Do They Stack Up?.* Retrieved July 18th 2007 from the O'Reilly publication website: http://www.oreillynet.com/pub/a/oreilly/java/news/farley_0800.html

[5] David Reilly. (n.d.). *The Java Programming Language.* Retrieved July 18 2007, from http://www.javacoffeebreak.com/articles/inside_java/insidejava-nov99.html

[6] William Crawford and Jonathon Kaplan. (2003). *J2EE Design Patterns.* O'Reilly

[7] Bill Siggelkow. (2005). *Jakarta Struts Cookbook*. O'Reilly

[8] Rod Johnson. (2007). *Spring 2.0: What's New and Why it Matters.* Retrieved July 18th, from http://www.infoq.com/articles/spring-2-intro

[9] Xavier Coulon and Christian Brousseau. (2004). *Hibernate simplifies inheritance mapping*. Retrieved July 18th, from http://www.ibm.com/developerworks/java/library/j-hibernate/

[10] Shine Technologies. (n.d.). *Real World Experiences with Hibernate.* Retrieved July 21st, from http://shinetech.com/pages/viewpage.action?pageId=649

[11] Thornton Rose and David Thurmond. (n.d.). *Jumping into JBoss.* Retrieved June 1st, from http://www.developer.com/java/ejb/article.php/3071661

[12] Qusay H. Mahmoud. (2004). *Developing Web Applications with JavaServer Faces.* Retrieved June 12th, from http://java.sun.com/developer/technicalArticles/GUI/JavaServerFaces/

[13] Laurance Moroney. (2005). *ICEfaces Offers a Novel, Pure Java Approach to the Rich vs. Thin Dilemma.* Retrieved July 2nd, from http://www.devx.com/webdev/Article/29068

[14] Kyle Gabhart. (2002). *JAAS Security in Action.* Retrieved July 3rd, from http://www.devx.com/getHelpOn/Article/9915/0

[15] Derek Yang Shen. (2004). *Put JSF to work.* Retrieved July 4th, from http://www.javaworld.com/javaworld/jw-07-2004/jw-0719-jsf.html

[16] Joshua D. Drake. (2005). *Open Source Database Feature Comparison Matrix.* Retrieved July 10th, from http://www.devx.com/dbzone/Article/29480

[17] The Server Side. (n.d.) *Application Server Matrix*, Retrieved July 20th, from http://www.theserverside.com/tt/articles/article.tss?l=ServerMatrix

[18] Jim Rapoza. (2004). *JBoss AS 4.0.* Retrieved July 21st, from http://www.eweek.com/article2/0,1759,1668400,00.asp

[19] J. Donovan Wells. (n.d.). *Extreme Programming: A Gentle Introduction.* Retrieved August 3rd 2007 from http://www.extremeprogramming.org/

[20] Ronald E Jeffries (2001). *What is Extreme Programming,* Retrieved July 21st, from http://www.xprogramming.com/xpmag/whatisxp.htm

[21] Steve Renaker (n.d.). *MySQL: A Lot More Going for It Than No Price Tag,* Retrieved July 21st from http://archive.devx.com/java/free/articles/Renaker02/Renaker02-1.asp

[22] Apache (2006). *Apache Ant,* Retrieved July 21st from http://ant.apache.org/

[23] IceFaces (2007). *Component Suite ShowCase,* Retrieved July 13th from http://component-showcase.icefaces.org/component-showcase/#

[24] Miller and Collins (2001). *XP Distilled,* Retrieved July 15th from http://www.ibm.com/developerworks/java/library/j-xp/?ca=drs-

[25] Kent Beck (2004). *Extreme Programming Explained: Embrace Change, Second Edition*. Addison Wesley Professional.

[26] Rational (1999). *Extreme Programming Practices.* Retrieved July 10th from http://ootips.org/xp.html

[27] Media Art online (2007). *How Java Works.* Retrieved July 11th from http://www.media-art-online.org/java/help/how-it-works.html

[28] Springframework (2007). *Spring Framework, An Introduction.* Retrieved July 8th from http://www.springframework.org/docs/reference/introduction.html

[29] JavaBeat (2007). *Hibernate Architecture.* Retrieved July 30th from http://hibernate.javabeat.net/tutorials/2-hibernate-architecture.php

Appendix A – Database Scripts

```
DROP DATABASE IF EXISTS exam;
CREATE DATABASE exam;
USE exam;

DROP TABLE IF EXISTS users;
CREATE TABLE users(
        username        VARCHAR(64) NOT NULL PRIMARY KEY,
        passwd          VARCHAR(64) DEFAULT NULL,
        first_name      VARCHAR(50) NOT NULL,
        last_name       VARCHAR(50) NOT NULL,
        address         VARCHAR(200) NOT NULL,
        phone           VARCHAR(50) NOT NULL,
        mobile          VARCHAR(50) NOT NULL,
        email           VARCHAR(50) NOT NULL,
        active          VARCHAR(5) NOT NULL DEFAULT 'false',
        last_login      DATETIME DEFAULT NULL,
        last_pwd_change         DATETIME DEFAULT NULL
)ENGINE=INNODB;



DROP TABLE IF EXISTS roles;
CREATE TABLE roles(
        userroles       VARCHAR(50) NOT NULL PRIMARY KEY,
        active          VARCHAR(5) NOT NULL DEFAULT 'false'
)ENGINE=INNODB;


insert into roles values('Super','true');
insert into roles values('Admin','true');


DROP TABLE IF EXISTS userroles;
CREATE TABLE userroles(
        username        VARCHAR(64) NOT NULL,
        userroles       VARCHAR(50) NOT NULL,
        PRIMARY KEY (username, userroles),
        FOREIGN KEY (username) REFERENCES users(username),
        FOREIGN KEY (userroles) REFERENCES roles(userroles)
)ENGINE=INNODB;


DROP TABLE IF EXISTS views;
CREATE TABLE views(
        view_name       VARCHAR(64) NOT NULL PRIMARY KEY,
        title           VARCHAR(100),
   active               VARCHAR(5) NOT NULL DEFAULT 'false'
)ENGINE=INNODB;


insert into views values('Main','Main View','true');

DROP TABLE IF EXISTS userviews;
CREATE TABLE userviews(
        username        VARCHAR(64) NOT NULL,
        view_name       VARCHAR(64) NOT NULL,
        PRIMARY KEY (username, view_name),
        FOREIGN KEY (username) REFERENCES users(username),
        FOREIGN KEY (view_name) REFERENCES views(view_name)
)ENGINE=INNODB;


--insert into users
values('super',AES_ENCRYPT('sup3r','fhjuioekjdknb'),'Super','User','No
Address','000000','NA','NA','Y',null,null);
insert into users values('super','sup3r','Super','User','No
Address','000000','NA','NA','Y',null,null);
insert into users values('snake','sup3r','Snake','User','No
Address','000000','NA','NA','Y',null,null);
insert into users values('donkey','sup3r','Donkey','User','No
Address','000000','NA','NA','Y',null,null);

insert into userviews values('super','Main');
insert into userviews values('snake','Main');
insert into userviews values('donkey','Main');


insert into userroles values('super','Super');
insert into userroles values('super','Admin');

insert into userroles values('snake','Super');
insert into userroles values('donkey','Super');


DROP TABLE IF EXISTS exams;
CREATE TABLE exams(
        exam_name       VARCHAR(64) NOT NULL PRIMARY KEY,
        view_name       VARCHAR(64) NOT NULL,
        date_created    TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        active          VARCHAR(5) NOT NULL DEFAULT 'false',
        FOREIGN KEY (view_name) REFERENCES views(view_name)
)ENGINE=INNODB;

-- insert into exams values('Java','Main',null,'true');
```

```
DROP TABLE IF EXISTS questions;
CREATE TABLE questions(
        q_id            INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        exam_name       VARCHAR(64) NOT NULL,
        question        VARCHAR(400) NOT NULL,
        difficulty      VARCHAR(50) NOT NULL,
        type            VARCHAR(50) NOT NULL,
        active          VARCHAR(5) NOT NULL DEFAULT 'false',
        username        VARCHAR(64) NOT NULL,
        FOREIGN KEY (username) REFERENCES users(username),
        FOREIGN KEY (exam_name) REFERENCES exams(exam_name)
)ENGINE=INNODB;

-- insert into questions values(1,'Java','This is the
question?','Easy','SingleAnswer','true','Super');
-- insert into questions values(2,'Java','This is the question
2?','Easy','MultiChoice','true','Super');
-- insert into questions values(3,'Java','This is the question
3?','Easy','SingleAnswer','true','Super');
-- insert into questions values(4,'Java','This is the question
4?','Easy','SingleAnswer','true','Super');

-- insert into questions values(5,'Java','This is the question
Med?','Medium','SingleAnswer','true','Super');
-- insert into questions values(6,'Java','This is the question Med
2?','Medium','MultiChoice','true','Super');
-- insert into questions values(7,'Java','This is the question Med
3?','Medium','SingleAnswer','true','Super');
-- insert into questions values(8,'Java','This is the question Med
4?','Medium','MultiChoice','true','Super');

-- insert into questions values(9,'Java','This is the question
Hard?','Hard','SingleAnswer','true','Super');
-- insert into questions values(10,'Java','This is the question Hard
2?','Hard','MultiChoice','true','Super');
-- insert into questions values(11,'Java','This is the question Hard
3?','Hard','MultiChoice','true','Super');
-- insert into questions values(12,'Java','This is the question Hard
4?','Hard','SingleAnswer','true','Super');



DROP TABLE IF EXISTS answers;
CREATE TABLE answers(
        a_id            INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        q_id            INT NOT NULL,
        answer          VARCHAR(400) NOT NULL,
        correct         VARCHAR(5) NOT NULL DEFAULT 'false',
        FOREIGN KEY (q_id) REFERENCES questions(q_id)
)ENGINE=INNODB;
```

```
-- insert into answers values(null,1,'answers ','false');
-- insert into answers values(null,1,'answers correct 2','true');
-- insert into answers values(null,1,'answers 3','false');

-- insert into answers values(null,2,'answers correct 4','true');
-- insert into answers values(null,2,'answers correct 5','true');
-- insert into answers values(null,2,'answers 6','false');

-- insert into answers values(null,3,'answers 7','false');
-- insert into answers values(null,3,'answers correct 8','true');
-- insert into answers values(null,3,'answers 9','false');

-- insert into answers values(null,4,'answers 10','false');
-- insert into answers values(null,4,'answers correct 12','true');
-- insert into answers values(null,4,'answers 13','false');

-- insert into answers values(null,5,'answers 14','false');
-- insert into answers values(null,5,'answers correct 15','true');
-- insert into answers values(null,5,'answers 16','false');

-- insert into answers values(null,6,'answers 17','false');
-- insert into answers values(null,6,'answers correct 18','true');
-- insert into answers values(null,6,'answers 19','false');

-- insert into answers values(null,7,'answers 20','false');
-- insert into answers values(null,7,'answers correct 21','true');
-- insert into answers values(null,7,'answers 22','false');

-- insert into answers values(null,8,'answers 23','false');
-- insert into answers values(null,8,'answers correct 24','true');
-- insert into answers values(null,8,'answers 25','false');

-- insert into answers values(null,9,'answers 26','false');
-- insert into answers values(null,9,'answers correct 27','true');
-- insert into answers values(null,9,'answers 28','false');

-- insert into answers values(null,10,'answers 29','false');
-- insert into answers values(null,10,'answers correct 30','true');
-- insert into answers values(null,10,'answers 31','false');

-- insert into answers values(null,11,'answers 32','false');
-- insert into answers values(null,11,'answers correct 33','true');
-- insert into answers values(null,11,'answers 34','false');

-- insert into answers values(null,12,'answers 35','false');
-- insert into answers values(null,12,'answers correct 36','true');
-- insert into answers values(null,12,'answers 37','false');

DROP TABLE IF EXISTS candidate;
```

```
CREATE TABLE candidate(
   username    VARCHAR(64) NOT NULL PRIMARY KEY,
   email               VARCHAR(64) NOT NULL,
   password    VARCHAR(64) NOT NULL,
   time        INT NOT NULL,
        noquestions     INT NOT NULL,
   view_name   VARCHAR(64) NOT NULL,
   active              VARCHAR(5) NOT NULL DEFAULT 'false',
   role        VARCHAR(9) DEFAULT 'ExamTaker',
        FOREIGN KEY (view_name) REFERENCES views(view_name)
)ENGINE=INNODB;

insert into candidate
values('Super1','sdfs@dsds.dd','noelk',5,5,'Main','true','ExamTaker');


DROP TABLE IF EXISTS candidateprofile;
CREATE TABLE candidateprofile(
        username        VARCHAR(64) NOT NULL,
        exam_name       VARCHAR(64) NOT NULL,
        FOREIGN KEY (username) REFERENCES candidate(username),
        FOREIGN KEY (exam_name) REFERENCES exams(exam_name),
        PRIMARY KEY (username, exam_name)
)ENGINE=INNODB;

--insert into candidateprofile values('Super1','Java');

CREATE OR REPLACE VIEW userpassview AS
        SELECT username AS uname, passwd AS pass FROM users
        UNION ALL
        SELECT username AS uname, password AS pass FROM candidate;

CREATE OR REPLACE VIEW rolesview AS
        SELECT username AS uname, userroles AS roles FROM userroles
        UNION ALL
        SELECT username AS uname, role AS roles FROM candidate;


DROP TABLE IF EXISTS candidateexams;
CREATE TABLE candidateexams(
   ce_id               INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
        username        VARCHAR(64) NOT NULL,
        starttime       TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP,
        endTime         TIMESTAMP NULL,
        FOREIGN KEY (username) REFERENCES candidate(username)
)ENGINE=INNODB;
```

```
DROP TABLE IF EXISTS candidatequestions;
CREATE TABLE candidatequestions(
   cq_id                    INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
         ce_id       INT NOT NULL,
         q_id              INT NOT NULL,
         questionNum  INT NOT NULL,
         FOREIGN KEY (ce_id) REFERENCES candidateexams(ce_id),
         FOREIGN KEY (q_id) REFERENCES questions(q_id)
)ENGINE=INNODB;

DROP TABLE IF EXISTS candidateanswers;
CREATE TABLE candidateanswers(
         cq_id             INT NOT NULL,
         a_id              INT NOT NULL,
   answer                   VARCHAR(5)  NOT NULL DEFAULT 'false',
   PRIMARY KEY (cq_id, a_id),
   FOREIGN KEY (cq_id) REFERENCES candidatequestions(cq_id)
)ENGINE=INNODB;
```

Appendix B – Sample Object Relational Mappings

Users Table

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated Feb 17, 2007 4:59:49 PM by Hibernate Tools 3.1.0.beta5 -->
<hibernate-mapping>
  <class name="ie.devplex.exam.model.Users" table="users" catalog="exam">
    <comment></comment>
    <id name="username" type="string">
      <column name="username" length="64" />
      <generator class="assigned" />
    </id>
    <property name="passwd" type="string">
      <column name="passwd" length="64">
        <comment></comment>
      </column>
    </property>
    <property name="firstName" type="string">
      <column name="first_name" length="50" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="lastName" type="string">
      <column name="last_name" length="50" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="address" type="string">
      <column name="address" length="200" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="phone" type="string">
      <column name="phone" length="50" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="mobile" type="string">
      <column name="mobile" length="50" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="email" type="string">
      <column name="email" length="50" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="active" type="string">
      <column name="active" length="5" not-null="true">
        <comment></comment>
      </column>
    </property>
    <property name="lastLogin" type="timestamp">
      <column name="last_login" length="19">
        <comment></comment>
      </column>
    </property>
    <property name="lastPwdChange" type="timestamp">
      <column name="last_pwd_change" length="19">
        <comment></comment>
      </column>
    </property>
    <set name="userviewses" lazy="false" cascade="all-delete-orphan"
inverse="true">
      <key>
        <column name="username" length="64" not-null="true">
          <comment></comment>
        </column>
      </key>
      <one-to-many class="ie.devplex.exam.model.Userviews" />
    </set>
    <set name="userroleses" lazy="false" cascade="all-delete-orphan"
inverse="true">
      <key>
        <column name="username" length="64" not-null="true">
          <comment></comment>
        </column>
      </key>
      <one-to-many class="ie.devplex.exam.model.Userroles" />
    </set>
    <set name="questionses" cascade="all-delete-orphan" inverse="true">
      <key>
        <column name="username" length="64" not-null="true">
          <comment></comment>
        </column>
      </key>
      <one-to-many class="ie.devplex.exam.model.Questions" />
    </set>
  </class>
</hibernate-mapping>
```

Roles Table

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated Feb 17, 2007 4:59:49 PM by Hibernate Tools 3.1.0.beta5 -->
<hibernate-mapping>
  <class name="ie.devplex.exam.model.Roles" table="roles" catalog="exam">
    <comment></comment>
    <id name="userroles" type="string">
      <column name="userroles" length="50" />
      <generator class="assigned" />
    </id>
    <property name="active" type="string">
      <column name="active" length="5" not-null="true">
        <comment></comment>
      </column>
    </property>
    <set name="userroleses" inverse="true">
      <key>
        <column name="userroles" length="50" not-null="true">
          <comment></comment>
        </column>
      </key>
      <one-to-many class="ie.devplex.exam.model.Userroles" />
    </set>
  </class>
</hibernate-mapping>
```

Appendix C – Spring Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
        <!-- ========================= Start of PERSISTENCE
DEFINITIONS ========================= -->

        <!-- DataSource Definition -->
        <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
                <property name="driverClassName">
                        <value>com.mysql.jdbc.Driver</value>
                </property>
                <property name="url">
                        <value>jdbc:mysql://localhost:3306/exam</value>
                </property>
                <property name="username">
                        <value>examuser</value>
                </property>
                <property name="password">
                        <value>exampass</value>
                </property>
        </bean>

        <!-- Hibernate SessionFactory Definition -->
        <bean id="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
                <property name="mappingResources">
                        <list>
                                <value>ie/devplex/bo/Test.hbm.xml</value>
                           <value>ie/devplex/exam/model/Users.hbm.xml</value>
          <value>ie/devplex/exam/model/Roles.hbm.xml</value>
          <value>ie/devplex/exam/model/Userroles.hbm.xml</value>
          <value>ie/devplex/exam/model/Userviews.hbm.xml</value>
          <value>ie/devplex/exam/model/Views.hbm.xml</value>
          <value>ie/devplex/exam/model/Exams.hbm.xml</value>
          <value>ie/devplex/exam/model/Questions.hbm.xml</value>
          <value>ie/devplex/exam/model/Answers.hbm.xml</value>
          <value>ie/devplex/exam/model/Candidate.hbm.xml</value>
          <value>ie/devplex/exam/model/Candidateprofile.hbm.xml</value>
          <value>ie/devplex/exam/model/Candidatequestions.hbm.xml</value>
          <value>ie/devplex/exam/model/Candidateexams.hbm.xml</value>
          <value>ie/devplex/exam/model/Candidateanswers.hbm.xml</value>
                        </list>
                </property>

                <property name="hibernateProperties">
                        <props>
                                <!-- <prop
key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop> -->
                                <prop
key="hibernate.dialect">org.hibernate.dialect.MySQLInnoDBDialect</prop>
                                <prop key="hibernate.show_sql">true</prop>
                                <prop
key="hibernate.cglib.use_reflection_optimizer">true</prop>
                                <prop
key="hibernate.cache.provider_class">org.hibernate.cache.HashtableCacheProvider</
prop>
                        </props>
                </property>

                <property name="dataSource">
                        <ref bean="dataSource"/>
                </property>
        </bean>

        <!-- Spring Data Access Exception Translator Defintion -->
        <bean id="jdbcExceptionTranslator"
class="org.springframework.jdbc.support.SQLErrorCodeSQLExceptionTranslator">
                <property name="dataSource"><ref bean="dataSource"/></property>
        </bean>

        <!-- Hibernate Template Defintion -->
        <bean id="hibernateTemplate"
class="org.springframework.orm.hibernate3.HibernateTemplate">
                <property name="sessionFactory"><ref
bean="sessionFactory"/></property>
                <property name="jdbcExceptionTranslator"><ref
bean="jdbcExceptionTranslator"/></property>
        </bean>

        <!-- Catalog DAO Definition: Hibernate implementation -->

        <bean id="testDAO"
class="ie.devplex.dao.hibernate.TestDAOHibernateImpl">
                <property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
        </bean>

        <bean id="usersDAO"
class="ie.devplex.dao.hibernate.UsersDAOHibernateImpl">
                <property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
        </bean>
```

```
	<bean id="rolesDAO"
class="ie.devplex.dao.hibernate.RolesDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>

	<bean id="viewsDAO"
class="ie.devplex.exam.dao.hibernate.ViewsDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>

	<bean id="examsDAO"
class="ie.devplex.exam.dao.hibernate.ExamsDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>

	<bean id="questionsDAO"
class="ie.devplex.exam.dao.hibernate.QuestionsDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>

	<bean id="userrolesDAO"
class="ie.devplex.dao.hibernate.UserrolesDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>

	<bean id="candidatesDAO"
class="ie.devplex.exam.dao.hibernate.CandidatesDAOHibernateImpl">
		<property name="hibernateTemplate"><ref
bean="hibernateTemplate"/></property>
	</bean>



	<!-- ========================= Start of SERVICE DEFINITIONS
========================= -->

	<!-- Hibernate Transaction Manager Definition -->

	<bean id="transactionManager"
class="org.springframework.orm.hibernate3.HibernateTransactionManager">
		<property name="sessionFactory"><ref
local="sessionFactory"/></property>
	</bean>
```

```
        <!--  User Service Defintion -->


        <bean id="testService" class="ie.devplex.service.impl.TestServiceImpl">
        <property name="testDAO"><ref local="testDAO"/></property>
        </bean>

        <bean id="userService"
class="ie.devplex.exam.service.impl.UserServiceImpl">
        <property name="usersDAO"><ref local="usersDAO"/></property>
        <property name="rolesDAO"><ref local="rolesDAO"/></property>
        <property name="viewsDAO"><ref local="viewsDAO"/></property>
        <property name="candidatesDAO"><ref
local="candidatesDAO"/></property>
        </bean>

        <bean id="examService"
class="ie.devplex.exam.service.impl.ExamServiceImpl">
        <property name="usersDAO"><ref local="usersDAO"/></property>
        <property name="rolesDAO"><ref local="rolesDAO"/></property>
        <property name="viewsDAO"><ref local="viewsDAO"/></property>
        <property name="examsDAO"><ref local="examsDAO"/></property>
        <property name="questionsDAO"><ref local="questionsDAO"/></property>
        <property name="candidatesDAO"><ref
local="candidatesDAO"/></property>
        </bean>


        <bean id="adminService"
class="ie.devplex.exam.service.impl.AdminServiceImpl">
        <property name="usersDAO"><ref local="usersDAO"/></property>
        <property name="rolesDAO"><ref local="rolesDAO"/></property>
        <property name="viewsDAO"><ref local="viewsDAO"/></property>
        </bean>



</beans>
```

Appendix D – Database Import

```
insert into questions values(1,'Sybase','Sybase Provide a Language to
enable you to communicate and manipulate objects in the ASE
environment. This language is called -
','Easy','SingleAnswer','true','Super');
insert into answers values(1,'1','JDBC','false');
insert into answers values(2,'1','ANSI SQL','false');
insert into answers values(3,'1','Transact SQL ','true');
insert into answers values(4,'1','ODBC','false');
insert into questions values(2,'Sybase','Which of the following does
not belong to the group.','Easy','SingleAnswer','true','Super');
insert into answers values(5,'2','master ','false');
insert into answers values(6,'2','system','true');
insert into answers values(7,'2','sybsystemprocs','false');
insert into answers values(8,'2','model','false');
insert into questions values(3,'Sybase','Put in the Correct Word. ASE
is a _______ Database Management
System','Easy','SingleAnswer','true','Super');
insert into answers values(9,'3','Distributed ','false');
insert into answers values(10,'3','Object Orientated','false');
insert into answers values(11,'3','Hierarchial','false');
insert into answers values(12,'3','Relational','true');
insert into questions values(4,'Sybase','Which of these statements is
false?','Easy','SingleAnswer','true','Super');
insert into answers values(13,'4','tempdb is recreated every time
Sybase ASE is booted','false');
insert into answers values(14,'4','tempdb should be as big as your
largest Database','true');
insert into answers values(15,'4','tempdb Performance should be
optimised by placing it on faster disk','false');
insert into answers values(16,'4','tempdb is only used for temporary
Processing ','false');
insert into questions values(5,'Sybase','Identify the False
Statement','Easy','SingleAnswer','true','Super');
insert into answers values(17,'5','Database Thresholds are used to
Monitor Free Space in any DB','false');
insert into answers values(18,'5','The Last Chance Threshold (LCT)
always exists in every user database.','false');
insert into answers values(19,'5','Users can create ThresholdAcions
to automatically trigger an event once threshold is
crossed.','false');
insert into answers values(20,'5','Database Thresholds must be
recreated everytime you reboot ASE','true');
insert into questions values(6,'Sybase','The Unit of Space Sybase
Allocates at  Database Creation
is','Easy','SingleAnswer','true','Super');
insert into answers values(21,'6','Bytes','false');
insert into answers values(22,'6','Blocks','false');
insert into answers values(23,'6','Pages','true');
insert into answers values(24,'6','Extents','false');
insert into questions values(7,'Sybase','The Minimum size of a User
Created Database must  always be at
least','Medium','SingleAnswer','true','Super');
insert into answers values(25,'7','512KB','false');
insert into answers values(26,'7','The Size of the Model
Database','true');
insert into answers values(27,'7','The size of tempdb','false');
insert into answers values(28,'7','The Size of the
sybsystemprocsdb','false');
```

```
insert into questions values(8,'Sybase','Select the Correct Statement
','Medium','SingleAnswer','true','Super');
insert into answers values(29,'8','Truncate can be used to delete a
sub-set of rows in a table','true');
insert into answers values(30,'8','Truncate Table is a  logged
operation','false');
insert into answers values(31,'8','You can make an incremental dump
(transaction dump) on a  DB where  a table has been
truncated.','false');
insert into answers values(32,'8','Truncate is always faster than
delete','false');
insert into questions values(9,'Sybase','Which 2 of the following can
tell you when your ASE server was
started?','Medium','SingleAnswer','true','Super');
insert into answers values(33,'9','Sybase Error Log','true');
insert into answers values(34,'9','select @@boottime','false');
insert into answers values(35,'9','sysservers table in Master
Database','false');
insert into answers values(36,'9','sysmonitors table in Master
Database ','false');
insert into questions values(10,'Sybase','To run the Server in Single
User Mode which option is passed to the RUN Server File.
','Medium','SingleAnswer','true','Super');
insert into answers values(37,'10','-p','false');
insert into answers values(38,'10','?s','false');
insert into answers values(39,'10','?m','true');
insert into answers values(40,'10','?i','false');
insert into questions values(11,'Sybase','Identify the true
statement','Medium','SingleAnswer','true','Super');
insert into answers values(41,'11','Sybase device sizes cannot exceed
32GB','true');
insert into answers values(42,'11','Sybase device sizes can be as
large as the underlying UNIX File System (UFS)','false');
insert into answers values(43,'11','Sybase device sizes are set &
controlled in the configuration files','false');
insert into answers values(44,'11','There is no limit to sizing a
Sybase Device','false');
insert into questions values(12,'Sybase','What is the maximum number
of Clustered Indexes that can be created on a
table?','Medium','SingleAnswer','true','Super');
insert into answers values(45,'12','1','false');
insert into answers values(46,'12','No maximum , depends on the
number of columns in the table','true');
insert into answers values(47,'12','Depends on the number of columns
/ one CI can be created for each column in the table','false');
insert into answers values(48,'12','256','false');
insert into questions values(13,'Sybase','Which statement does not
effect attaining  FAST BCP ','Hard','SingleAnswer','true','Super');
insert into answers values(49,'13','Where there are Clustered Indexes
on the Table','false');
insert into answers values(50,'13','Where there are Non-Clustered
Indexes on the Table','false');
insert into answers values(51,'13','Where no database options are
set','false');
insert into answers values(52,'13','Where the Data contains Character
& Integer ','true');
insert into questions values(14,'Sybase','Stored Procedures : Which
statement is False?','Hard','SingleAnswer','true','Super');
insert into answers values(53,'14','Stored Procedures are compiled
once and its Query Plan is stored in memory  (Procedure
cache)','false');
```

```
insert into answers values(54,'14','A stored Procedure can be used to
call another Stored Procedure in the same database.','true');
insert into answers values(55,'14','Sybase Provided Stored Procedures
are  always prefixed with sp_   (  e.g. sp_helpdb)','false');
insert into answers values(56,'14','Stored Procedures can use tables
in another database on the same server.','false');
insert into questions values(15,'Sybase','Engines : Which of the
following statements is
False?','Hard','SingleAnswer','true','Super');
insert into answers values(57,'15','The number of Active Engines
cannot be greater than the number of normalised CPU?s','false');
insert into answers values(58,'15','There must always be at least two
engines active, 1 for  dataserver and one for backup
server.','true');
insert into answers values(59,'15','The number of engines is defined
in the cfg file','false');
insert into answers values(60,'15','There must be at least 2 engines
to accommodate parallel processing ','false');
insert into questions values(16,'Sybase','Triggers : Which of the
Statements is False?','Hard','SingleAnswer','true','Super');
insert into answers values(61,'16','A Trigger is a Stored Procedure
that goes into effect when you INSERT,DELETE or UPDATE data in a
Table','false');
insert into answers values(62,'16','Triggers can cascade changes
through related tables in the database ','false');
insert into answers values(63,'16','Triggers can be used to create
object in the database','true');
insert into answers values(64,'16','Trigger Execution is
automatic.','false');
insert into questions values(17,'Sybase','Dump & Recovery
Which of these statements is true?
','Hard','SingleAnswer','true','Super');
insert into answers values(65,'17','The Size of a Full  Dump without
compression is the total DB Size ','false');
insert into answers values(66,'17','The Size of the Full Dump without
compression will be equal to the amount of data space used by the
database ','true');
insert into answers values(67,'17','The size of the dump includes the
total size of Data & Transaction Logs','false');
insert into answers values(68,'17',' Always the size of the device(s)
on which the DB resides ','false');
insert into questions values(18,'Sybase','Which of the following
explanation  is  a correct explanation of the  SQL syntax ?
   isnull(expression_1,expression_2)
','Hard','SingleAnswer','true','Super');
insert into answers values(69,'18','Compares the two values
expression_1 and expressoion2 and if they are the same returns a NULL
Value.','false');
insert into answers values(70,'18','if expression_1=expression_2,
returns null','false');
insert into answers values(71,'18','Substitutes the value in
expression_2 when expression_1 is NULL otherwise returns
expression_1','true');
insert into answers values(72,'18','Returns the value of the first
expression in the list which is not null','false');
insert into questions values(19,'Solaris','Which command is used to
find free space per file
system','Easy','SingleAnswer','true','Super');
insert into answers values(73,'19','df -k','true');
insert into answers values(74,'19','free -k','false');
insert into answers values(75,'19','dfree -k','false');
insert into answers values(76,'19','freespace -k','false');
insert into questions values(20,'Solaris','Which command is used to
search through a file called filename for a string "search
string"','Easy','SingleAnswer','true','Super');
insert into answers values(77,'20','string "search string"
filename','false');
insert into answers values(78,'20','find "search string"
filename','false');
insert into answers values(79,'20','grep "search string" filename
','true');
insert into answers values(80,'20','find filename "search string"
','false');
insert into questions values(21,'Solaris','Which command is used to
unmount  a CD-ROM and eject
it','Easy','SingleAnswer','true','Super');
insert into answers values(81,'21','eject cd-rom','false');
insert into answers values(82,'21','eject cdrom','true');
insert into answers values(83,'21','eject ?m cd','false');
insert into answers values(84,'21','eject /cdrom/cdrom0','false');
insert into questions values(22,'Solaris','Which command is used
view disks in Solaris','Easy','SingleAnswer','true','Super');
insert into answers values(85,'22','format','true');
insert into answers values(86,'22','diskview','false');
insert into answers values(87,'22','showdisks','false');
insert into answers values(88,'22','disklist -v','false');
insert into questions values(23,'Solaris','Which file has a list of
file systems to mount at
boot','Medium','SingleAnswer','true','Super');
insert into answers values(89,'23','/etc/vfstab','true');
insert into answers values(90,'23','/etc/defaults/fs','false');
insert into answers values(91,'23','/etc/mnttab','false');
insert into answers values(92,'23','/etc/autofs','false');
insert into questions values(24,'Solaris','Which port and protocol
does telnet use','Medium','SingleAnswer','true','Super');
insert into answers values(93,'24','21 / UDP','false');
insert into answers values(94,'24','21 / TCP','false');
insert into answers values(95,'24','23 / UDP','false');
insert into answers values(96,'24','23 / TCP','true');
insert into questions values(25,'Solaris','Which file contains one or
more share commands used with the shareall
command','Medium','SingleAnswer','true','Super');
insert into answers values(97,'25','/etc/nfs/nfstab','false');
insert into answers values(98,'25','/etc/nfs/shares','false');
insert into answers values(99,'25','/etc/dfs/dfstab','true');
insert into answers values(100,'25','/etc/dfs/sharetab','false');
insert into questions values(26,'Solaris','Which command is used to
view system calls in a running process with pid
123','Medium','SingleAnswer','true','Super');
insert into answers values(101,'26','coreadm ?p 123','false');
insert into answers values(102,'26','truss ?p 123','true');
insert into answers values(103,'26','sysview ?p 123','false');
insert into answers values(104,'26','dump 123','false');
insert into questions values(27,'Solaris','Which unix command can be
used to modify an NVRAM
parameter','Medium','SingleAnswer','true','Super');
insert into answers values(105,'27','eeprom','true');
insert into answers values(106,'27','setnvram','false');
insert into answers values(107,'27','setprom','false');
insert into answers values(108,'27','setenv','false');
insert into questions values(28,'Solaris','Which command will replace
string "from" with string "to" in file
/test','Medium','SingleAnswer','true','Super');
insert into answers values(109,'28','sed ''s/from/to/g''
/test','true');
insert into answers values(110,'28','sed ''s/from/to/g'' ?f
/test','false');
insert into answers values(111,'28','sed ''r/from/to/''
/test','false');
insert into answers values(112,'28','sed ''r/from/to/g''
/test','false');
insert into questions values(29,'Solaris','Which of the following
files match the metacharacter
[abc][123]','Medium','SingleAnswer','true','Super');
insert into answers values(113,'29','abc123','false');
insert into answers values(114,'29','b2','true');
insert into answers values(115,'29','ab2','false');
insert into answers values(116,'29','ab3','false');
insert into questions values(30,'Solaris','Which command is used to
manage dynamic configuration of hot-plug
devices','Medium','SingleAnswer','true','Super');
insert into answers values(117,'30','cfgadm','true');
insert into answers values(118,'30','dynadm','false');
insert into answers values(119,'30','hpadmin','false');
insert into answers values(120,'30','attach','false');
insert into questions values(31,'Solaris','Which command is used to
find the current run-level','Medium','SingleAnswer','true','Super');
insert into answers values(121,'31','runlevel -c','false');
insert into answers values(122,'31','runlevel','false');
insert into answers values(123,'31','who -runlevel','false');
insert into answers values(124,'31','who -r  ','true');
insert into questions values(32,'Solaris','Which command is used to
view disks in veritas ','Hard','SingleAnswer','true','Super');
insert into answers values(125,'32','vxdisk list','true');
insert into answers values(126,'32','vxdiskinfo -l','false');
insert into answers values(127,'32','vxdsk -info ','false');
insert into answers values(128,'32','vxformat','false');
insert into questions values(33,'Solaris','Which crontab executes  on
Sunday at 02:10','Hard','SingleAnswer','true','Super');
insert into answers values(129,'33','2 10 * * 0
/var/tmp/check.sh','false');
insert into answers values(130,'33','10 2  * * 0
/var/tmp/check.sh','true');
insert into answers values(131,'33','2 10 * * *
/var/tmp/check.sh','false');
insert into answers values(132,'33','10 2 * * *
/var/tmp/check.sh','false');
insert into questions values(34,'Solaris','Which command is used to
find out about shared memory ','Hard','SingleAnswer','true','Super');
insert into answers values(133,'34','sminfo','false');
insert into answers values(134,'34','lsmem','false');
insert into answers values(135,'34','mem -s','false');
insert into answers values(136,'34','ipcs','true');
insert into questions values(35,'Solaris','Which command is used to
find out what modules the kernel has loaded
','Hard','SingleAnswer','true','Super');
insert into answers values(137,'35','modules -i','false');
insert into answers values(138,'35','modinfo','true');
insert into answers values(139,'35','kerninfo','false');
insert into answers values(140,'35','modkern -L','false');
insert into questions values(36,'Solaris','What command is used to
look for open file descriptors for a process named test with pid
123','Hard','SingleAnswer','true','Super');
insert into answers values(141,'36','fdescript 123','false');
insert into answers values(142,'36','ofiles ?p 123','false');
insert into answers values(143,'36','ofiles 123','false');
insert into answers values(144,'36','pfiles 123','true');
insert into questions values(37,'Solaris','A user wants to see what
shared files are mounted remotely and where they have been mounted.
What file would display this
information?','Medium','SingleAnswer','true','Super');
insert into answers values(145,'37','/etc/dfs/dfstab','false');
insert into answers values(146,'37','/etc/hosts','false');
insert into answers values(147,'37','/etc/nsswitch.conf','false');
insert into answers values(148,'37','/etc/rmtab','true');
```