Woodgate Software: High End Architecture and Prototype

Ross Woodgate  
CPIT  
Madras Street  
Christchurch 8011  
+64 3 9408888  
rww0065@student.cpit.ac.nz

Alison Clear  
EIT, Auckland  
Airedale Street  
Auckland 1010  
+64 9 4444303  
aclear@eit.ac.nz

ABSTRACT
One of the authors has been involved with Accounting and Business Management software for 15 years and has a vision for designing his own Business Management with a focus later on job costing systems. This project looks at the design of some of the initial high end architecture and building a prototype. In particular the login routine that needs to look at many databases and variables to determine a valid entry into the system. There is also a complicated hierarchy to the settings and security features that are required due to the system being built to suit many customers’ requirements. The asynchronous messaging required to inform all users of what records are required by other users and to allow them to act on this information was developed. The project also looks at a standard saving form for customers and also a transactional entry form. Three main modules were developed and fully tested.

Categories and Subject Descriptors
D.2 SOFTWARE ENGINEERING

General Terms
Management, Documentation, Design,

Keywords
Accounting software, database, software engineering

1. INTRODUCTION
This project focused on the set up the Front End GUI, Back End Web Service, and databases required for the development of the initial high end architecture and a prototype for the development of business accounting software.

This project built the logic for:
- How we connect to the Databases
- Communication between front end and back end applications
- Logging On
- Building the menu structure
- Standard ‘Master Table’ Form
- Standard Transactional Form

2. DEVELOPMENT OF THE PROJECT
2.1 Structure of the Program
The application was designed so any front end device will be able to be set up easily. The frontend can be any mobile device, a website or a PC App and can communicate with the web service successfully. The structure also enables the software owner to manage licensing requirements with one cloud License Database, enabling the implementation of temporary licensing in the future.

2.2 PostgresSql and C#
For this project it was decided to program in C# and to use PostgresSql as the database. C# was chosen due to this being the most popular programming language for applications and the
ease of writing in C#. PostgreSQL was chosen primarily because it has no limits to size and is an open source free ware database, it is also a database that the programmer has used previously.

2.3 Logon procedure
When the login screen is loaded it looks at the Master Database for the list of companies available and loads the company dropdown box with the list of companies
It then checks in the licensing database
- The number of licences for the current logged on
- the Licence Database that the user logged in has access to the relevant company
- the number of licences currently in use is less than number of licences
- If licence available then we insert a new record into companyuseraccess
It then checks in the Company Data Table
- If user exists (note double check user should always be in both the Master and Licence Database)
- If password correct
- Then it creates a session and returns the sessionID back to the login form and loads the menu

2.4 Building the Menu
The menu was designed to be as wide as the screen and to have an area to show any messages sent to the particular session. This will allow for use of Excel or other packages to be used easily without the software taking any more real estate than necessary.

2.5 Record Locking
Due to the fact we were using a disconnected database record locking was required to be developed it was decided to use sessionID as the record locking method. When a user logs on they create a session record in the customer database. When we need to get exclusive access to an item eg editing a record then we update the sessionID field on that record with the current sessionID. If someone else requires access the method of looking at the lock will be as follows:
- If there is no sessionID against the record then we can use safely
- We then look at the Session record to see if the session has been logged off if it has we remove the sessionID from the record we want to edit and continue on.
- If there is a sessionID and the session is not logged off we see how long the session has been idle by looking at the session record dateLastTrans if this is older than a preset time then we will through Asynchronous messaging communicate with the session that has the record locked,
- If no confirmation message is returned from the front end then we will assume the session has been disconnected and will log off the session remove the record lock we know about and continue on
- If a confirmation record has come back we will advise user the record is a valid lock and a message has been sent to the session and we are awaiting further action

2.6 Nunit Testing
Nunit Testing was setup to check the routines and that they worked ok. This was not totally completed the framework for testing was developed and setup.
What this requires is setting assertion messages in the test regime and running the test when it returns an assertion that does not match the programed assertion message it returns an error to the Nunit testing platform.

3. PERSONAL REFLECTION
This project has taught me quite a bit to what is required to complete the software. I have struggled at many corners to understand the concepts required to complete this project.
Some of the many difficult concepts I have learnt from this project are:
- Web Service programing
- Asynchronous Messaging
- Decryption /Encryption
- Nunit Testing
- Passing variables and classes to front end using binary serialisation

4. CONCLUSION
The three main modules of the proposed Business and Accounting Management software were fully developed and tested. The Front End GUI, Back End Web Service, and databases required for the development of the initial high end architecture and a prototype for the development of business accounting software were completed.

5. REFERENCES