2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Investigating the Suitability of a .NET/SQL Server Approach for Developing a Remotely Accessible Information System for Vehicle Inspectors

Justin Kumar [1], Zenon Chaczko [2], Bruce Moulton [3] , Venkatesh Mahadevan [4 +]

[1, 2 & 3] University of Technology Sydney, Australia

[4] Swinburne University of Technology, Australia

Abstract. Vehicle inspectors using current systems typically make mental or brief written notes while inspecting a vehicle, to be entered into a computer system at a later date. We propose a paperless system for vehicle inspection, and investigate some software engineering tools and methods for its development. The work was initiated as part of a final year computer system engineering thesis project. The research project involved analyses of the requirements and functional specifications, and included the design, implementation and analysis of a working prototype system. Preliminary evaluation of the approach suggests that it appears to be suitable for the development of the specified vehicle inspection information system. The approach is intended to be scalable, but questions remain as to the extent to which the approach is suitable for the development of a larger scale deployment.

Keywords: NET, SQL Server, C#, and User Interface.

1. Introduction

There are approximately 45,000 vehicle accidents in New South Wales (NSW), Australia each year. Each vehicle that is repaired requires an inspection from an insurance or damage assessor. The current assessment process typically involves an inspector taking mental or written notes of the damage, sometimes making a written notes, and passing those notes to a third person (distant from the vehicle and inspection process) or entering them onto a computer at a later point in time. The current system leaves room for error due to imperfect recollection on the part of the assessor, or misinterpretation of the inspection notes by data entry personnel. In addition, many vehicle inspection systems do not permit inspection records to be stored or accessed from a remote location, and rarely have the functionality to store images. t was proposed that .NET and SQL Server tools may be suitable for the development of a system to enable paperless processes for capturing, storing and accessing inspection data. This paper describes the development of such as system. t was thought that the benefits of the system might include elimination of the manual paper to computer data entry process, increased productivity and accuracy, savings due to reduced double handling and data entry, and benefits due to inspection data being available for immediate processing.

Thus work was undertaken to investigate the suitability of .NET and SQL Server tools for the development of a remotely accessible vehicle inspection information system. This work began as part of a final year computer system engineering thesis project. The project was of particular interest to the student because the student has first hand involvement with vehicle inspections in the automotive industry. It was thought that even though the problem space might be unique, the challenges associated with the development of the system might be similar to those of other systems.

[+] Corresponding author. Tel.: + 61 3 92157315; fax: + 61 3 92154713.
E-mail address: vmahadevan@swinburne.edu.au.

It was proposed that vehicle inspectors might make use of a mobile device such as a PDA while performing inspections, and that this method could provide a streamlined approach for capturing and storing inspection data and negate some of the limitations of traditional inspection processes.

2. Method, Architecture and Design

The prototype system uses Bluetooth to wirelessly transfer data between each mobile device and the server, Although Bluetooth was the chosen wireless communication technology for the prototype system, it is acknowledged that other communication technologies such as GPRS and WiFi may be preferable in many deployments. Each mobile device runs a SQL Server database, as does the remote server. Merge replication is used to synchronise data between the databases. The remote server creates a template of its database and sends it to the wireless device's compact database. The remote server then creates a snapshot of key data from its database and sends it to the wireless device's compact database. Changes can be made to the snapshot on the mobile device and the snapshot on the remote server. These individual snapshots are then merged when a synchronisation process is actioned.

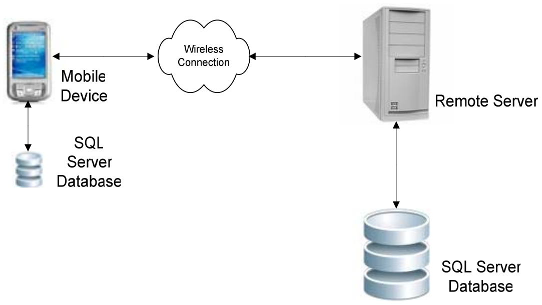

Fig. 1: Component diagram of the prototype system.

The documentation and design of the project was informed by reference to the following standards.

- The SDP document is based upon IEEE STD 1058-1998.
- The SRS is based upon the 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- The SDD is based upon the IEEE STD 1016-1998
- The STP is based upon the 829-1983 (R1991) IEEE Standard for Software Test Documentation.

Several development tools were used to develop the prototype system. The primary tools are described in the following Table 1 and the prototype system's deployment component descriptions are given in Table 2.

Table. 1: Technologies used for development and design

Development tool	Role
MS Visual Studio 2008	Integrated Development Environment
SQL server 2008	Relational database management system
SQL Server Compact Edition 3.5	Version of SQL Server suitable for mobile devices
.NET Framework 3.5	Handle interoperability issues between applications
.NET Compact Framework 3.5	Version of .NET Framework suitable for mobile devices

Table. 2: Application deployment component roles.

Component	Role
User Interface	Provides an interface between the user and the system.
Job Management	Manages job related activities including the searching for, viewing, updating, deleting and creating of job records
Access Management	Manages the access to the system. Access activities include logging into and out of the system.
Administration	This component manages the account maintenance activities that are performed by system administrators.
Database	This component stores the system's data.

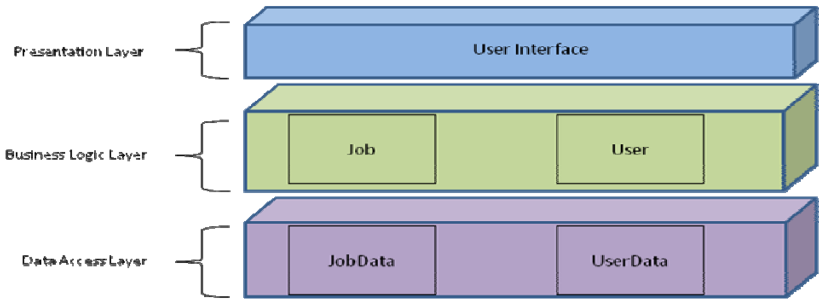

Fig. 2: Architecture.

The prototype system includes two applications, one for the mobile device and another for the desktop. Both are structured using a three-tier architecture design, as illustrated in Figure 2. The advantages of this architecture include (a) scalability–the system can be easily scaled in each layer with minimal impact on the processes within adjacent layers; (b) improved security–firewalls can be placed between each layer increasing the number of security checks required by the system; and (c) database server specialisation–separating the database allows the database server to be more specialised.

The system has two primary user groups, system administrators and mechanics (or inspectors). The system administrators are able to set up user accounts and access all aspects of the system. The system allows the inspectors to record the details of inspections or jobs on a mobile device, at the time they perform the inspection, as opposed to taking a mental note of the job and then later recording it. The system is self contained and whole; it does not require communication from any external systems and is not part of a larger system.

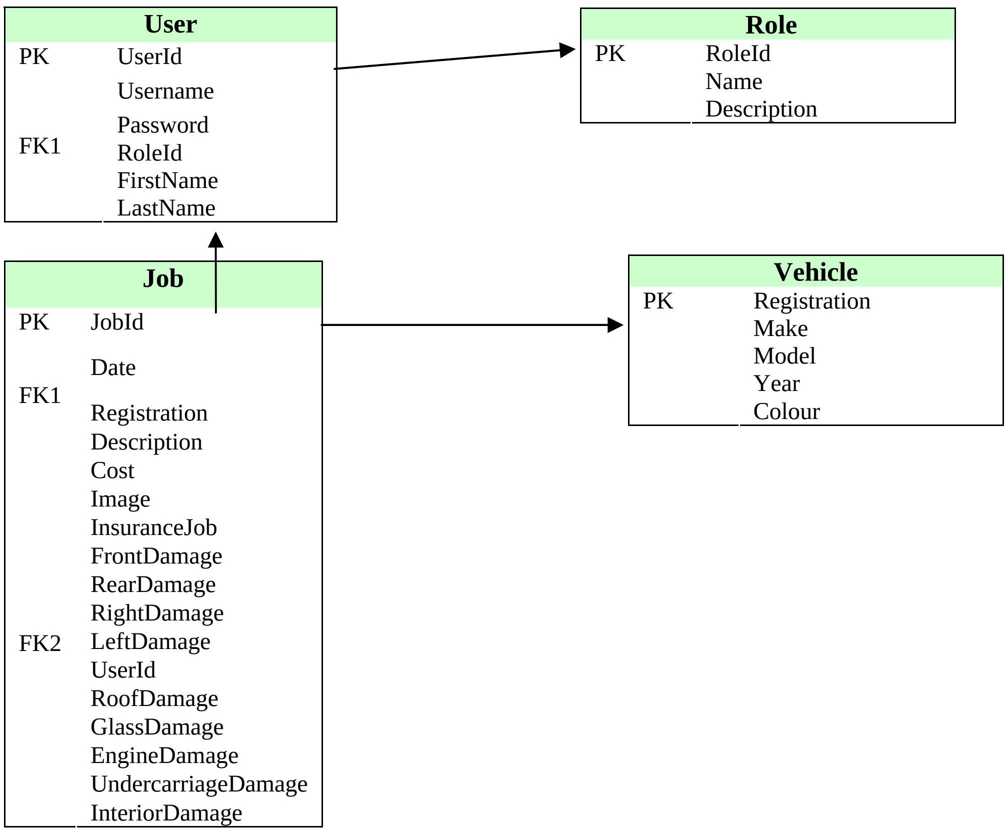

Fig. 3: Data base mapping.

System users interface with the system through the system application installed on a wireless mobile device. The application is developed in C #. The mobile device runs Windows Mobile 6.0. Data from the mobile device is transmitted to a remote server which in the prototype is represented by a Personal Computer

(PC) equipped with Bluetooth. The data transmitted by the mobile device is stored in a database. The database edition used for the prototype system is Microsoft SQL Server 2008. Database mapping is shown in Figure 3.

2.1. Analysis of Prototype System Functions

The application’s main menu options were intended to vary according to the role of the logged in user. Only administrators have the ability to access account maintenance areas. The applications allow users to perform functions including the following (note that not all functions are listed here):

- Login: the user is able to log into the system using a user name and password.
- Recording a job: users can create a new inspection job. After requesting to create a new job, the system displays mandatory fields which the user fills with relevant data.
- Saving a job: after creating a job and filling all mandatory fields, the user can save the job details (and send job for storage/analysis).
- Creating/altering/deleting a user account: system administrators have the ability to create new user accounts and maintain existing accounts.
- Printing: inspection job details may be printed
- Capturing an image: users can capture and store images using a mobile device.

Testing indicated that the prototype system performs all of the required functions.

2.2. Analysis of Prototype User Interface

The user interface was analysed by constructing flow diagrams. Examples include the user interface (UI) flow for the desktop application, which is shown in Figure 4, and the UI flow for the mobile application, along with the mobile application job search window, which is shown in Figure 5. Preliminary analyses indicate that the UI appears to be suitable for the meeting the required specifications.

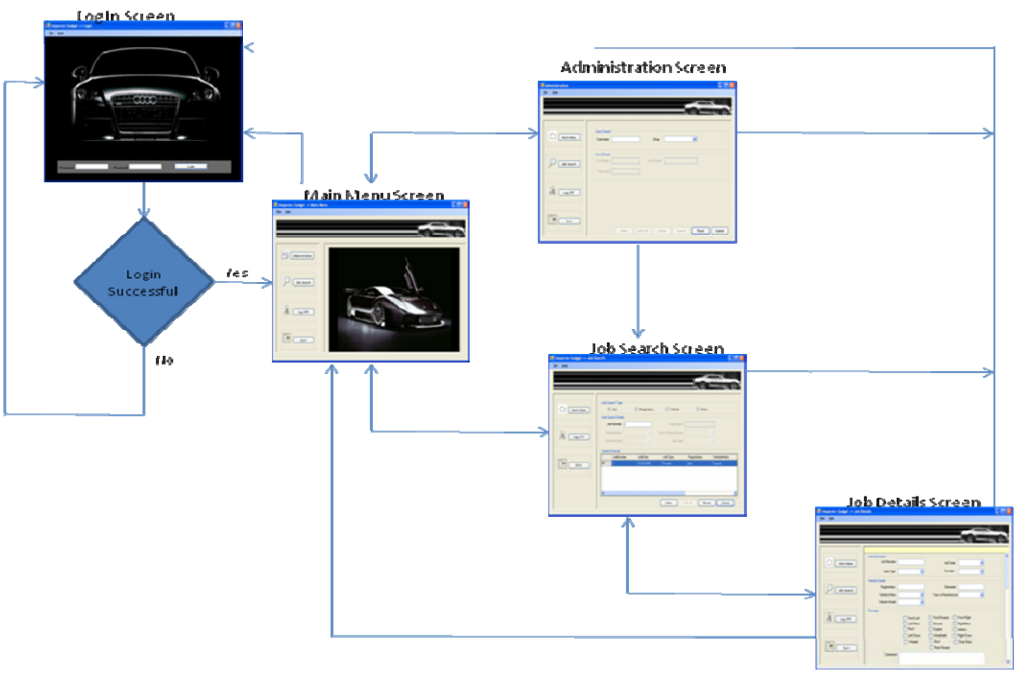

Fig. 4: UI object desktop user interface flow.

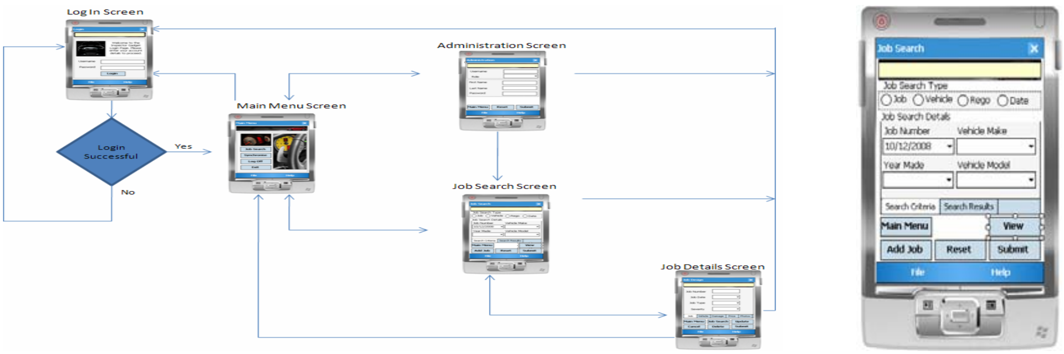

Fig. 5: UI object mobile user interface flow and UI object mobile job search window.

3. Preliminary Evaluation of System Development

All requirements in the SRS were tested along with all user interface rules specified in the UI Specifications. Unit testing focused on the individual modules of the program. Its primary purpose was to determine whether or not modules are correctly coded and working as design. Acceptance testing was based on the rules within SRS, primarily to assess the extent to which the system met the requirements and is operational. The output of each test was documented, and all outstanding defects after the testing phase were recorded in an error log within the software test report document. Two overall results emerged from the preliminary evaluation. First, enabling an inspector to record data and images at the time of the inspection appears to be advantageous, especially given that an inspector is not obliged to use the system at the time of the inspection if she prefers not to. (In that case, the system might simply be used as a faster way to store/organise relevant images.) Second, the tools and approach appear to be suitable for the development of the specified vehicle inspection information system. The choice of tools and the approach was developed with scalability as one of the foremost considerations. Even so, it is acknowledged that scaling up a system such as this is likely to raise unforeseen issues, hence this is an area where further work is required. However, prior research which relates to vehicle inspection in general includes studies of the effectiveness of vehicle inspections, for example, [1] and [2] and sensor/robotic systems for vehicle inspection, for example, [3]. During the project we became aware of commercially available software that can be used by vehicle inspectors on a PDA [4]. Also of interest to this project is research undertaken on a wearable computer that may assist vehicle inspectors focusing on issues associated with non-standard dial-type UI device [5].

4. Conclusion

The analysis suggests that the approach and tools were quite suitable for the development of the prototype system. The system is designed to be scalable, but due to the complexity of issues that arise when attempting to scale up a system, questions remain as to whether the approach is indeed suitable for larger scale deployment.

5. References

[1] Pierson, W.R. (1996) Motor vehicle inspection and maintenance programs—How effective are they? *Atmospheric Environment* Volume 30 21: i-iii.

[2] Merrell, D., Poitras, M. and Sutter, D. (1999) The effectiveness of vehicle safety inspections: An analysis using panel data. *Southern Economic Journal*, 65 3: 571-583.

[3] Page, D., Y. Fougerolle, A. Koschan, A. Gribok, M Abidi, D. Gorsich, and G. Gerhart, "SAFER Vehicle Inspection: A Multimodal Robotic Sensing Platform," *in Proc. SPIE Unmanned Ground Vehicle Technology VI*, Vol. 5422, Orlando, FL, USA, pp. 549-560, April 2004.

[4] http://www.pdspect.com/vehicle/ [accessed 29.4.2009].

[5] Bass, L., Kasabach, C., Martin, R., Siewiorek, D., Smailagic, A., and Stivoric, J. 1997. The design of a wearable computer. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, March 22 - 27, 1997).