

Simple Mobile Warehouse System for Microsoft Dynamics Navision ERP System

Vladimír Gašpar^{1,*}, Ladislav Madarász¹, Ján Paralič¹, Katarína Ténaiová², Jozef Vido²,
Zuzana Grančáková², Martin Riňak²

¹Department of Cybernetics and Artificial Intelligence, Technical University of Košice, Košice, 042 00, Slovak Republic

²Department of implementation of ERP systems, ICOS, a.s. Košice, Košice, 040 01, Slovak Republic

Abstract The paper describes a methodology for creating a simple modular warehouse system involving novel structures and modifications of existing mobile frameworks. The novelty of the presented approach is the platform independence and simplicity of the architecture, simple prototype of a GUI and platform independent layer communication model. The proposed methodology is illustrated on the example of a real model application which was designed for a private company. The system described and created is intended for use over Dynamics Navision ERP system. A lot of these topics may be considered challenging in the future research of business application development in general, either for desktop, web or mobile platforms. Since prospects in the field of mobile applications and mobile business solutions (m-business) are really promising for all leading PDA/tablet platforms, there is an expectation of a rising demand in the next years.

Keywords Warehouse System, PDA, Framework, Layer Model, Business Applications, Effectiveness, User Interface, ERP

1. Introduction

The latest information research concentrates on the performance of designed software applications. The performance of an application heavily depends on its software structure and capabilities of hardware platform. Every structure has its specificity and is suited for different operating systems or hardware configurations. This fact can be clearly seen in large information systems. For example, the possibility of creating applications for ERP systems using efficient software structures is fairly low because ERP systems are usually built for a particular system configuration. However, the ERP systems are considered essential for the information effectiveness in a company[5-7]. Enhanced access to relevant information can improve the company's processes.

Nowadays, mobile solutions are becoming popular and can deliver required information at any place and any time[12]. However, developers never gave much attention to the mobile application development for ERP systems. Only few mobile frameworks for ERP systems were officially released. Moreover, these frameworks are not platform independent.

We need to mention that only few simple and ERP sys

tem specific applications have been created so far. There is already an application “@NAV - Mobile Warehouse by bartolome röder AG” but this application does not consider the structural software aspect. Therefore it is not optimized for particular load requirements[11]. Moreover, it is difficult to carry out possible platform changes. The purpose of this application is rather general. Our novel approach also uses available frameworks but we have considered the structural aspect and proposed efficient software structures suitable for different load levels and customer requirements. The latest importance of examined topics are discussed in various publications[13-16].

For better information representation, a specific made-to-measure GUI was also created. We consider the further development of mobile solutions for ERP systems essential and also very promising. Since the mobile business applications market is not yet saturated, there is also a considerable space for new research and enhancements, whether in case of simple solutions for specific conditions or wide ranged universal solutions suitable for various systems[6].

2. Architecture and Background

2.1. Simple Model Possibility

The foundation of the proposed and created solution is mainly framework driven. It relies on the classic windows mobile framework .NET Compact Framework in 3.5 version and Microsoft Dynamics Mobile 1.5 (MDM) supplementary framework. This framework is used mainly for

* Corresponding author:

vladimir.gaspar@tuke.sk (Vladimir GAŠPAR)

Published online at <http://journal.sapub.org/se>

Copyright © 2012 Scientific & Academic Publishing. All Rights Reserved

designing because of its natural inclination to the simple graphical business processes management. These frameworks had been chosen for the best compatibility with ERP system for Windows. The MDM supports a wide range of integration schemes for particular customer demands, the size of the company, the number of warehouses operating within the company and a lot more. It also includes graphical components suitable for creating the entire visual application environment.

The main advantage of this framework is modification variability in accordance with requirements. When needed, the framework can be slimmed up to several essential components. This backbone components configuration is suitable for small scale systems or small companies with only few requests pending at a time. If a customer needs a powerful system with possibility of handling large numbers of requests, then the full architecture should be used. The framework modifications should be taken care of in the phase of implementation. The complete unchanged architecture provided by MDM is depicted in Fig. 1.

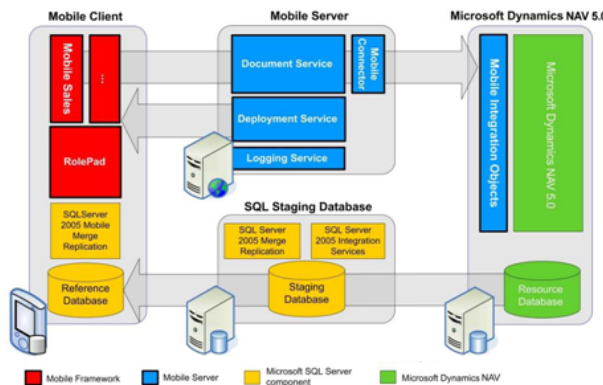


Figure 1. Basic, non-modified MDM structure[11]

This architecture configuration provides extended data handling, which enables large amounts of data with a low latency to be transferred. It is important to note that every group of elements may, but does not have to be, physically separated. When using separate physical servers for each group, one can achieve great horizontal scalability with a load balancing function involved. Moreover, when operating with many users, web services may be used for separating communication handling. The separation is represented by mobile integration components on the application server side and by mobile connector on the web services side. It is not possible to separate the request and the response in terms of the communication only but also the data flow between databases[8].

Complexity of the resource database (the native ERP system database) may be reduced using a staging database with integration services and the replication web service on the database server side. This fact decreases amount of data needed for the application to work offline with synchronized data. However, when small amounts of data are expected, the staging database is usually omitted. Finally, all essential data are replicated to the reference database, lo-

cated in the target mobile device. With Microsoft SQL Server, only the merge replication is allowed for the mobile database. It is handled by the Microsoft SQL Server CE database server. The replication process consists of[9]:

- creation of data publications on the server side,
- creation of data subscriptions on the client side,
- data synchronization.

Publications are represented by sets of permissions that allow the replication for subscribed users in some extent, but only if required conditions are met. The subscription in the client database has to be linked with a matching publication. Subscriptions can be either permanent, or they may have an expiration time. When it expires, a new one has to be created, or the following synchronizations fail and throw an exception[9].

The client side is independent and is not affected by the used architecture. The structure of the application is represented by a rolepad project, which executes every module created as a tasklet project. A tasklet defines one particular screen. These screens are usually logically attached to each other with input and output variables or objects defined in the UserRole.xml file. It is also possible to orchestrate tasklets and create an application according to existing workflows. There are two basic types of orchestrations[10]:

- hierarchical orchestration (Fig. 2),
- wizard orchestration (Fig. 3).

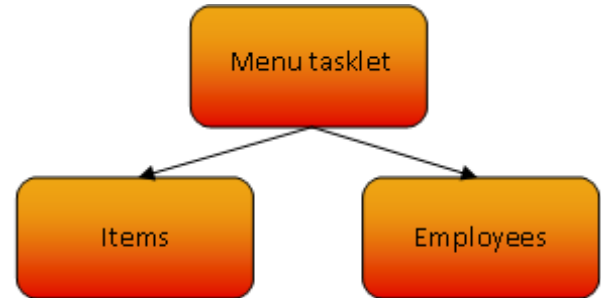


Figure 2. Hierarchical orchestration depiction



Figure 3. Wizard orchestration depiction

In the hierarchical type, tasklets in lower hierarchical level are accessed from those in the level above. The wizard type is typical for situations that require usage of several steps to complete the workflow. This means that tasklets are orchestrated in a particular order one after the other.

Special tasklets like the menu tasklet, are created to simplify development. These tasklets are not designed through the designer, they just use specific tags and their properties in previously mentioned UserRole.xml file. All other configurations, settings, service references, web service references and other connection strings are defined in a separate file, app.config.

2.2. Layered Model Possibility

In the previous chapter, we considered a simple model that only involved the framework elements. These were made by the same company (Microsoft). This fact should eliminate any incompatibilities.

When considering a layered model of the framework, we may divide the vertical dimension into (see. Fig. 4):

1. Client layer
2. Web services layer
3. Application layer
4. Database layer

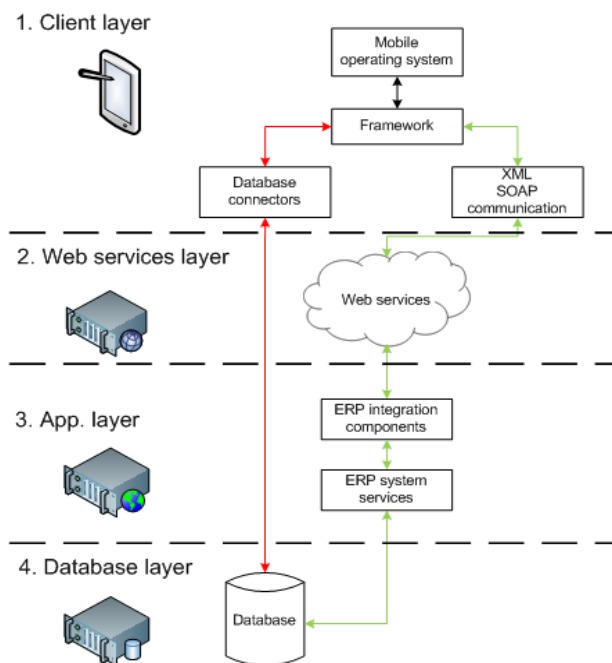


Figure 4. Layered model of the ERP system communication possibilities

The first layer serves as a presentation layer (front-end). It is obvious that this layer of the framework may be modified or ported to any other mobile operating system i.e. *Android*, *Windows Phone 7*. With the use of proper connector tools for databases (i.e. ODBC) and web services, which are considered a separate layer, the programmers are able to integrate any type of frontend framework into existing MDM structure.

The web service layer exchanges data, requests and responses in form of XML documents (via SOAP messages). This fact creates the opportunity to use the existing web services, when implementing different front-ends. Although web services themselves are suited for deployment on a windows based server system, the web service target operating system does not influence the rest of the framework (Fig. 4 – green path).

The application server layer is platform dependent and is only suitable for the particular ERP system (in our case the Dynamics Navision). However, there is a possibility of creating a scheme for each required ERP system database and consider direct access to the data without using layers 2 and 3. Depending on the detected or chosen ERP system

database scheme, the front-end behavior would change.

The database layer is mainly related to connector tools that we have already mentioned. The best solution is to use front-end that supports direct connection with the database through supported connectors (Fig. 4 – red path). Some unexpected issues may occur when trying to pair unsupported database with the chosen front-end. Moreover, many ERP systems (including Dynamics Navision) have a native type of database that may complicate the solution. The best compatibility is ensured when using widely used database engines with large support, i.e. *mySQL*, *MS SQL*, *postgreSQL*, *Oracle*, etc.

3. Analysis and Environment Description

The company has already have an existing computer network structure within its company network. It uses Microsoft Dynamics Navision 5.0 SP1 as its ERP system. The actual warehouse implementation in the ERP system does not differ from the demo company CRONUS - SLOVENSKO s.r.o. Processes for warehouse activities control are slow, and unnecessary preliminary paperwork is required. Warehouse processes have the following structure:

3.1. Goods Expenditure

- The warehouse worker prepares goods for the expenditure in the required place based on preliminary expenditure paper.
- The operator, based on an expenditure receipt, confirms the expenditure paper, which counts off the item from the warehouse.
- The warehouse worker hands over goods to the shipper, based on the confirmed expenditure paper.

3.2. Goods Intake

- The warehouse worker checks the physical amount of goods that will be positioned in the warehouse and notes it to preliminary intake paper.
- The warehouse operator confirms the intake and registers the intake paper.

3.3. Goods Transfer

- The warehouse worker checks the physical amount of goods in the warehouse and notes it to a transfer record.
- The warehouse operator confirms the transfer and registers the transfer record.

3.4. Physical Inventory

- An inventory template, in which the physical status of goods is being written, is printed continuously or regularly.
- If any differences are discovered, they are registered in the ERP system and compensated with the real status of goods.

The existing data flow in the company's warehouses network lacks wireless elements. This means that the network structure will need to be slightly modified during the implementation process to enable wireless connection for PDA devices.

As a default user, we consider any employee in any company warehouse and other terrain employees performing warehouse or inventory activities. The company requires an increase in employees' flexibility, a better view on warehouse items and a simplification of all previously mentioned warehouse processes.

4. Possible Simple Solutions

According to company's information system, different approaches to solve the problem are possible. Each of the possibilities is specific and requires the same data but different components and frameworks. They also have different advantages and disadvantages. In the following chapters we compare possible solutions only on one module (the warehouse items filter module).

4.1. SharePoint Services 3.0 Modules

Integrating the application into the SharePoint portal, or in case of the analyzed company the intranet, is a promising possibility from a basic point of view. The main advantages of this solution are:

- more filter settings can be displayed,
- web site is platform independent from the client side and no requirements except for a basic internet browser are required for the web site to work.

Figure 5. Items filter design as a SharePoint web part

Description	Measure 1	Measure 2	Measure 3	Location	Reason of measure
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS
Pa plach varenost na tople	0,012000/3000	00PL00020	0000	12000.00	XIS

Figure 6. View of the module integrated in the SharePoint portal[11]

However, the usage on mobile devices is rather questionable. PDA devices usually do not display the content of web pages that are optimized for desktop personal computers correctly. Since the SharePoint portal is optimized for higher resolutions and for desktop web browsers, the displayed content will be corrupted. Another possibility is to use a personal computer. The disadvantage in this case is the lack of mobility. When using a laptop, it is the size of a device that the warehouse worker has to carry in the warehouse. The possible design of this solution is depicted in Fig. 5 and Fig. 6.

4.2. ERP System Client Application Modules

The second possible solution is using the internal programming capabilities of the ERP system client to create a specific required module. For example in case of SAP we may use the *ABAP* programming language and in case of Dynamics Navision we may use the *C/SIDE* language.

The biggest disadvantage of such a solution is that the client application has to be physically installed on the host computer. This possibility is not supported by PDA devices at all. Moreover when connecting from remote locations, the user will have to use the VPN (requires high bandwidth internet connection) to connect to the internal company network at first.

The most considerable advantage is the compatibility and data integrity without any need of afterward checking. Basic validation functions and stored procedures take care of the data integrity after entering any new record or accepted warehouse operation. The design of this solution possibility integrated in *Dynamics Navision 5.0* using *C/SIDE* is depicted in Fig. 7.

The last chosen possibility is described in detail in chapter 5.

Figure 7. Items filter design as the Navision client module

5. Solution Proposal

Using PDAs, we can achieve company's requirements for the application. There would be no more need for the preliminary paperwork, the flexibility will increase with enhanced warehouse workers mobility. Also, using barcode scanners may improve intake, transfer, expenditure and physical inventory processes. The usage of the layered model will allow the scalability of the created application.

5.1. Network Modifications

It is necessary to add several wireless access points to the network structure of every warehouse to cover up the most of the working area. Even though minimum interference expected, it is possible that stored goods made of metal or ferro-concrete may cause some signal obstructions. This is why the final application has to be connection independent and able to work offline in case the connection to the access point is lost. Therefore, the local database with manually synchronized data directly from the ERP system is necessary. The proposed network topology is depicted in Fig. 8.

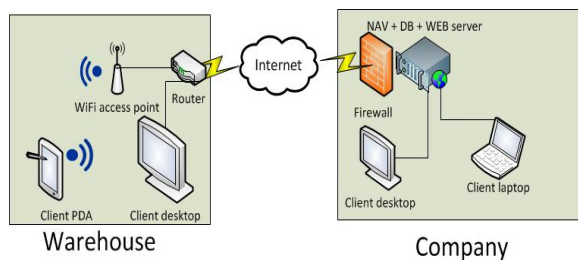


Figure 8. Proposed network topology[11]

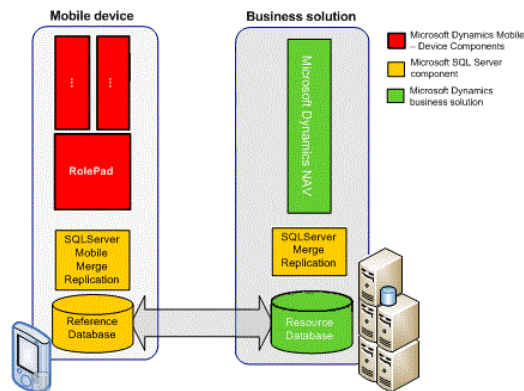


Figure 9. MDM modified architecture[11]

5.2. Framework Architecture Modifications

The MDM framework also had to be modified to simplify and speed up the data exchange (see Fig. 9). Using this particular architecture, only one-way server-to-device synchronization is utilized. The other way is presented by direct access to the resource database. This approach enables user's control over the data exchange at any time. While there is no need to exchange large amounts of data, the mobile staging database with integration services and also web services that handle requests and responses had been omitted.

For each of these omitted services we found a compensation solution.

In case of layered model, proposed in chapter 2.2, we consider using only layers 1 and 4. Since the target company is small, it is necessary to simplify the solution, optimize it for a small company and a small expected number of requests. This solution will lower the overhead in device-to-device communication, where the principal "lower the number of entities in communication, lower the communication latency" is held.

5.2.1. Document Service

When a direct access approach to the database is used, there is no need to use web services to handle the data writing to the database. All records are inserted directly by created SQL statements. This principle assures that the data will be inserted in a correct form with every necessary attribute and a correct date.

5.2.2. Logging Service

Since the logging service is primarily used with the document service, we consider the native SQL database logging file (*.ldf) associated with every Microsoft SQL Server database sufficient.

5.2.3. Deployment Service

The deployment service is not necessary, because there are not large numbers of devices used. The created application may be copied manually to each PDA device.

5.3. User Interface

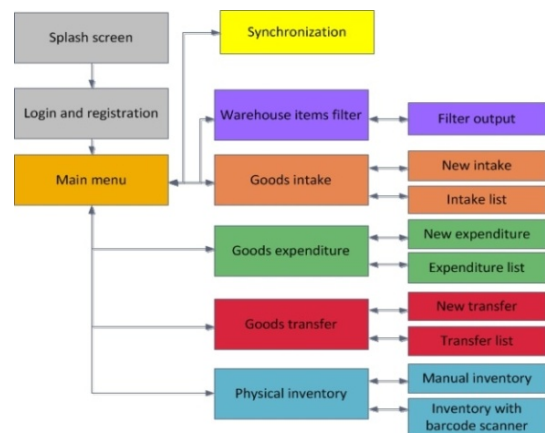


Figure 10. User interface modules diagram[11]

The user interface is divided into orchestrations. Every orchestration is depicted with a different color in Fig. 10. Arrow directions show possible transitions between orchestrations or tasklets. The application will start with a loading screen (splash screen). After the application is fully loaded, the login and the registration screen will appear. If login is successful, the user will be permitted to access the application's main menu. After logging in, the user will not be able to log in with different credentials, unless he shuts the application down. Every module will be accessible from the

main menu or a submenu of each warehouse activity. Using such a division of a user interface enables a good overview and a certain degree of elasticity when working with different warehouse activities.

5.4. Data Analysis

There are two main data storage places. Firstly, there is a resource database which contains data necessary for reading information about existing items, warehouses. All of these data are a subject of the replication, so that the mobile application is able to work offline. These tables are listed in Table 1.

Table 1. Database tables chosen for replication

NAV number	Table name	Description
27	Item	Items table
32	Item Ledger Entry	Items movements table
14	Location	Warehouse properties
5723	Product Group	Different product groups and their properties
5722	Item Category	Different item categories and their properties
5717	Item Cross Reference	Barcode - Item number references table

There are also tables which are needed for writing, and which contain information about created intakes, expenditures, transfers and inventory differences. The communication with these tables (i.e. Item Journal Line, Item Batch Line) will be directly implemented in SQL statements in the application's source code.

Then it is the reference database which contains information about all intakes, expenditures, transfers and inventory status created with the particular PDA device. It also stores login names and password hashes of registered users. We propose using the MD5 as the default hashing algorithm but it can be changed to any other hashing algorithm provided by the .NET Compact Framework 3.5. For security reasons, only hashes of passwords will be stored, so that passwords cannot be retrieved or changed. Access to the database will be granted through login credentials of the mobile application, since when a user tries to register, entered information is tested as login credentials to the SQL Server itself. Users are granted basic rights for reading, writing and updating particular tables in the ERP system. Users are not allowed to delete data from tables or drop these tables.

6. Created Application and its Properties

The created application consists of 16 projects, 14 of which are tasklet projects, one is a rolepad project and the other one is a class library project. The class library project stores global application settings, for example connection strings for databases, the company name, etc. Following figures are screenshots from the actual application that the

customer received after the end of application development. The application has been running on a high-end HTC HD 2 Leo PDA device.

6.1. Visual Functional Application Modules Description

In terms of the application flow (Fig. 10), the first screen user sees is the splash screen. It shows the status of the application loading, presented by the red status bar and a status text (see Fig. 11. - left). Next, there is the login and the registration form. To log in, the user has to fill in credentials and choose the warehouse he is currently working in (see Fig. 11. - right). There is also a possibility to omit the warehouse selection and work within all company's warehouses.



Figure 11. Splash screen and the login /registration form[11]



Figure 12. Main menu and synchronization screens[11]

The main menu is organized automatically using the default menu tasklet (see Fig. 12 - left). Every menu "action" tag with its "open" and "icon" properties, represents one menu icon. These action attributes are created and managed in the UserRole.xml file.

The last visual functional module is the synchronization. As it was previously stated, the synchronization consists of publications, subscriptions and the data synchronization process. When trying to synchronize, the PDA device has to create subscriptions for existing publications to the reference database. This process is shown on the screen of the PDA device. If subscriptions already exist, the synchroniza-

tion process is started. If any of required subscriptions do not exist, they are instantly created. After every required subscription has been added, the synchronization process starts. The competition and the actual synchronization state are shown on the PDA screen (see Fig. 12 - right).

6.2. Warehouse Functions Modules Description

Every created module is different and provides different functionalities. However, three modules are basically slight modifications of each other. Intake, expenditure and transfer of goods modules differ in source warehouse, target warehouse fields combinations and the target table in which every created record is stored.

6.2.1. Warehouse Items Filter Module

This module is useful for discovering amounts of goods in a particular warehouse. It consists of two tasklets. The first one is the screen with a set of filters (see. Fig. 13 - left). It enables a user to use any filter combination to find the exact item and view its properties. The second screen is the list of filtered items. This list contains several columns, each of which has been picked by the target company (see. Fig 13 - right).



Figure 13. Warehouse items filter[11]



Figure 14. New warehouse item movements screens[11]

This module uses notifications when the connection status changes. The system responds on the ActiveSync and the WiFi connection status. When the connection is active, the external database is used and when the connection is lost, the data source is switched to use the internal database.

6.2.2. Intake, Expenditure and Transfer of Goods Modules

As it was previously mentioned, these modules have only slight but important differences. Each of these modules contains a navigational submenu, which a user can access two basic functionalities from. The first one is for creating new goods movements (see. Fig. 14).

The second one is a list of already issued and confirmed movements with a binary flag denoting, whether the movement has already been synchronized (see Fig. 15 - left). If considering a small screen size, it is very important that some functions are shown only in pop-up windows or context menus. Functions for adding, deleting, editing items in an actual movements list, saving to the database, the barcode scanning and the sending to the ERP system are located in the context menu (see Fig. 15 - right). A user can call the context menu by holding a finger on the touch screen for 1-2 seconds.



Figure 15. New warehouse item movements screens[11]

6.2.3. Physical Inventory

Because barcode scanning enabled application was requested, we have created two separate functions for working with the physical inventory. The first function is a manual physical inventory (see Fig. 16. - left). Every amount discovered by manual counting has to be written in the textbox for the particular item, which is shown on the screen. After the last amount of the last item is entered and confirmed, the application prompts a user to finish and send the inventory to the ERP system. After the inventory is successfully sent, every record from the "physicalInventory" table in the local database is removed. The inventory can only be done for that warehouse which the user is currently logged in. If the user is not logged in a warehouse, a "%" sign will appear which means that every existing warehouse will be used (see Fig. 16. - right).

Also the physical inventory with the barcode scanning is possible and speeds up the inventory process. When the barcode is scanned, the application automatically finds the corresponding item and shows its properties. The user can choose whether he wants to add or remove the scanned item from the inventory count. After the inventory is finished, the user sends all records to the ERP system with one click on the diskette icon. If an inventory is not finished due to call or any other interruption, the user can continue or start over.

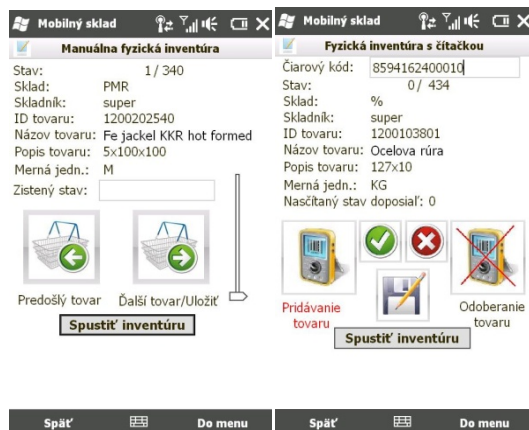


Figure 16. New warehouse item movements screens[11]

7. Conclusions

In the paper, we proposed a novel approach to mobile software development for business applications over the ERP systems. We proposed the platform independent model, and expressed possibilities and advantages of existing framework modifications. We also created a simple warehouse system for MS Dynamics NAV using proposed structures and framework modifications. The mobile application was created with respect to requirements of a private company. Moreover, we compared the behavior advantages and disadvantages of possible warehouse implementation in web and desktop environments. We assume that the application usage would bring benefits mainly to companies with small or medium sized warehouses. Proposed work also brings up the question of the multiplatform framework and possibilities of the application scalability.

ACKNOWLEDGEMENTS

The paper has been supported by the research project of VEGA 1/0298/12 “Digital control of complex systems with two degrees of freedom”.

REFERENCES

- [1] Tserng, P. - Dzeng, R.J. - Lin, Y. Ch. - Lin, S. T.: Mobile Construction Supply Chain Management Using PDA and Bar Codes, 2005, accessible on the internet: http://www.cv.nctu.edu.tw/~wwwadm/chinese/teacher/paper_teach23/Mobile%20Construction%20Supply%20Chain%20Management%20Using%20PDA%20and%20Bar%20Codes.pdf
- [2] Kornak, A. - Teutloff, J. - Welin-Berger, M.: Enterprise guide to Gaining Business Value from Mobile Technologies, 2004
- [3] Jelassi, T. - ENDERS, Albrecht: STRATEGIES for e-BUSINESS, Creating Value through Electronic and Mobile Commerce, 2004
- [4] Hladký, R.: Utility evaluation of company information systems mobile accessibility, 2004 Systems Integration, paper is accessible on the internet: <http://si.vse.cz/archive/proceedings/2004/hodnoceni-vyuzitelnosti-mobilniho-pristupu-k-podnikovym-informacnim-systemum.pdf> (in Czech)
- [5] Madarász, L. - Bučko, M. - Andoga, R.: Integrational aspects of creation and operation of CIM systems. Elfa, s.r.o., FEI TU Košice, 380pp. ISBN 80-8086-043-2, 2006.(in Slovak)
- [6] Madarász, L. - Bučko, M. - Andoga, R. - Gašpar, V.: System analysis and synthesis. Elfa s.r.o., FEI TU Košice, 2012, 303 pp., ISBN 978-80-8086-193-3. (in Slovak)
- [7] Basl, J. - Blažíček, R.: Company information systems. Company in information society. Grada, Praha, 2008, 283pp., ISBN 978-80-247-2279-5. (in Czech)
- [8] Microsoft Official Training Materials for Microsoft Dynamics NAV Mobile 2008™
- [9] Sujoy, P. P.: Pro SQL Server 2008 Replication, Apress 2009
- [10] Microsoft Dynamics® Mobile White Paper: Extending Business Solutions to the Mobile Workforce, Microsoft 2008
- [11] Gašpar, V. - Madarász, L. - Paralič, J. - Ténaiová, K.: Design and implementation of a client warehouse application over an enterprise resource planning system for mobile devices. 2011. 5 pp. In: proceedings IEEE conference LINDI 2011 Óbuda University Budapest, Hungary. ISBN 978-1-4577-1840-3.
- [12] Delina, R. - Vajda, V.: Theory and practice of electronic trading. GRAFOTLAČ PREŠOV, s.r.o., EkF TU Košice, 2008, 172 pp., ISBN 978-80-969953-3-2. (in Slovak)
- [13] Wasserman, Anthony L.: Software engineering issues for mobile application development, FoSER 2010.
- [14] Thompson, C. - White, J. - Dougherty, B. - Schmidt, D.: Optimizing mobile application performance with model-driven engineering In: Software Technologies for Embedded and Ubiquitous Systems., pp. 36-46, Heidelberg, Springer Verlag 2009. ISBN 978-3-642-10264-6.
- [15] Kim, J.: Architectural patterns for service-based mobile applications. In: proceedings of Service-Oriented Computing and Applications (SOCA), 2010 IEEE International Conference on Service-Oriented Computing and Applications., Perth, Australia, 2010. ISBN 978-1-4244-9802-4
- [16] Gavalas, D.: Development Platforms for Mobile Applications: Status and Trends, In: Software, IEEE vol. 28 issue 1. 77-86pp. 2011.