

# **ShopSYS: Management Software for Small Businesses**

**Done By: Soufiane Berouel**  
**Supervised By: Lily R. Liang**

**Department of Computer Science and Information Technology**  
**University of the District of Columbia**  
**Washington, District of Columbia, 20008, USA**  
**sberouel@udc.edu**

**2/06/2009**

## **Abstract**

In this paper we present a software system, ShopSYS, which we developed for small business management. As large businesses, mini shops also need automated computer software systems to make management efficient. However, their small scale requires especially simple and easy-to-use software that are tailored to their basic needs. ShopSYS integrates several functions that a mini shop would use for their daily operation which allows a manager or employee to supervise and run stock, cash flow, and transactions in their business. Managers can also oversee and keep record of their employees using ShopSYS. This project is done by two undergraduate computer science students as a course project for their Software Engineering class. In the development of this software, we applied the principles of software engineering to a simulated real-world scenario. This result was presented and demoed to the class.

## **1. INTRODUCTION**

As large businesses, mini shops also need automated computer software systems to make management efficient. However, their small scale requires especially simple and easy-to-use software that are tailored to their basic needs. ShopSYS, a software system that we developed, answers these needs of a small shop. It contains five major functionalities and two access levels. Through the security function, the manager will be able to access stock, employees' records, cash flow, and transactions. An employee however may access only stock and transactions. Manage stock and employee records are very similar since they both carry the five common functions in data handling which are add, delete, modify, search, and view. Manage cash flow allows the system manager to keep track of the monthly expenses and revenues and as a result find the benefits. Manage transaction contains the common functions in every shop business including bill processing and waiting lists management.

The project was created for software engineering class during the spring 2008 semester through a period of ten weeks. In the development of this software, we applied the principles of software engineering to a simulated real-world scenario. We followed the distinct phases of software development cycle, which we learned in the software engineering class which goes from studying the feasibility of the project to implementation and testing. The system application is created using NetBeans integrated development environment (IDE) and java programming language. Some other tools such as Photoshop and MS-Paint were also used to create diagrams and figures. This project was also a team work which required good communication, productive contribution, and subordination of individual aspirations to the efficiency of the group.

## **2. METHODOLOGY**

To develop ShopSYS, we went through six distinct and major phases using the waterfall model as a development process. As its name suggests, the waterfall model forces the engineer to pay great attention to every phase because there is no going back. We had to make sure not to rush moving to next stage until the current one is totally complete.

First of all, we conducted feasibility study to ensure that the software is feasible to be completed by the end of the semester with resources that were available. After that, we conducted a thorough investigation of which functions a small shop manager could require in a system application. These requirements were grouped together to generate a design of the system. We designed the architecture of the software system and a friendly graphical user interface (GUI) which will help the user get familiar with the product after only few tries. Then we moved to implementing the design of ShopSYS using java programming language and NetBeans IDE. We chose the incremental prototyping model for this phase so that we can make a prototype of the program and then add to it based on any new and necessary changes. Finally, we conducted multiple testing and debugging sessions to make sure that ShopSYS is fully functional and trust worthy.

The following table represents those phases with respect to time:

**Table 1: ShopSYS Time Table**

Period \ Phase	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
Feasibility Study										
Requirements Specification										
Architecture Design										
User Interface Design										
Software Implementation										
Testing										

### 2.1 Feasibility Study

The feasibility study is a general examination of the feasibility of an idea to be converted and executed within the constraints of times and resources. It was done in during the first week of ShopSYS development. Since ShopSYS is an in class project, no financial or contractual issues were involved thus financial resources were not a concern. Technological resources needed, e.g. software development tools and computers were available. However, time was a primary concern. In a preliminary draft we have concluded that the system could have probably more than forty functions. Even with past experiences in large projects using java, we knew it would be challenging to develop such as a big number of functions in a period of three to four weeks. At the end, we have decided to move forward since all functionalities seemed irremovable and necessary for a small shop to function.

### 2.2 Requirements Collection

After the conduction of the feasibility study which approved this project in terms of time and available resources, it was time for collecting the requirements. To do so, it was necessary to stop by a couple of mini shops in my area and get a look on their systems. This gave me a concrete idea of the main functionalities that the system should include.

The first and most important thing is security, all users of ShopSYS should have an account and managers can create an account for their employees. Every user should be equipped with a username and a password to access ShopSYS. The security function also specifies the layers of access for different types of users. There are functionalities that only managers can access. Namely, the manager can access and modify an employee's record and also view the total revenues of the shop. These functions are not available for an employee. The latter however is eligible to manage stock and approve selling transactions. Another measure of security deals with the copyrights of the program. To assure that only businesses that buy ShopSYS will be able to use, it was mandatory to create a key for system setup and a private key generator.

The management of cash flows in a business is quite a simple task, every month a new balance sheet is created. Whenever an item is bought and added to stock, the purchase price of the item is added to the expenses of that month. Whenever an item is sold, the selling price is added to the revenues. The benefits are computed with a simple subtraction operation and displayed to the manager.

Managing stock and employee records are very similar tasks. Therefore, only manage stock class will be developed in details. In managing stock, the user shall be able to add and view items as well as their specifications (selling price, provider information, quantity, etc). If a mistake is done, users are able to go back and modify the specifications of the item anytime they want. To do so, they should go back to stock and search for the item first using the search function that can be performed either by the name or code of the item. Finally, when stock almost runs out of an item, ShopSYS will launch an alert to remind the user of buying it. Except from the alert function, managing employees is exactly the same. Items and employees have different attributes like salary instead of selling price, age instead of quantity, etc. However, the functionalities involved in both tasks are not different at all.

The last and final function is the backbone of any shop management application. Managing transaction is where most of the time will be spent by users of ShopSYS. Managers or employees are able to create bills for costumers based on products they bought. These bills can be printed and a soft copy of a bill is always saved to certify that the costumer really bought an item. As transactions are made, the stock is automatically updated and sold items are removed. Also, the revenues and benefits of the corresponding month are updated too. To avoid a number of real life business complications, ShopSYS allows the creation of a waiting list of temporary bills since some costumers may need to stop the transaction and go get some other items or even forget to bring their wallets. Having the option of moving bills to waiting lists could help avoid stagnation in costumer lines which is not very efficient.

The entire functional requirement gathered above could be grouped and represented in a hierarchal manner as the Figure 1 illustrates.

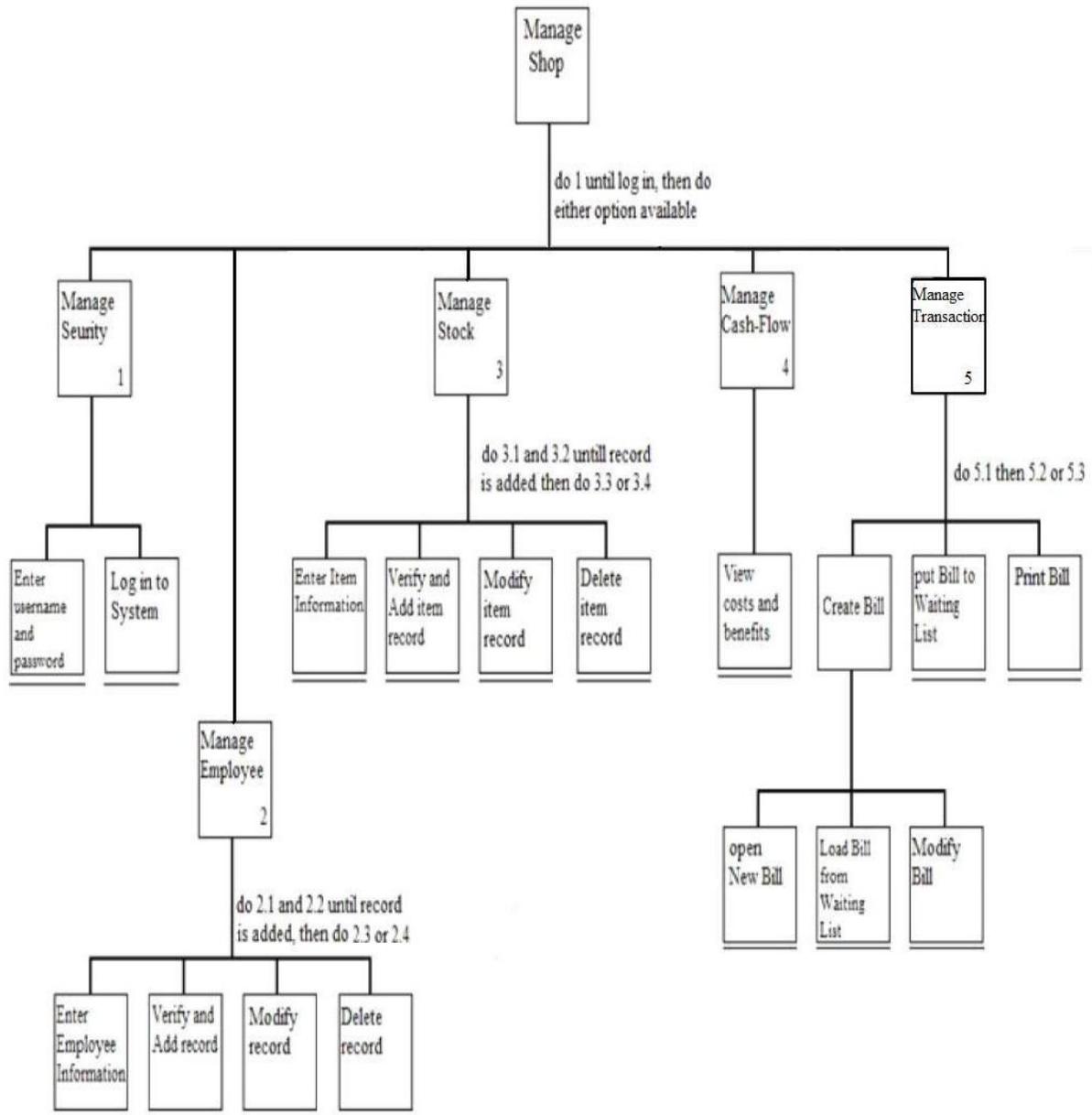
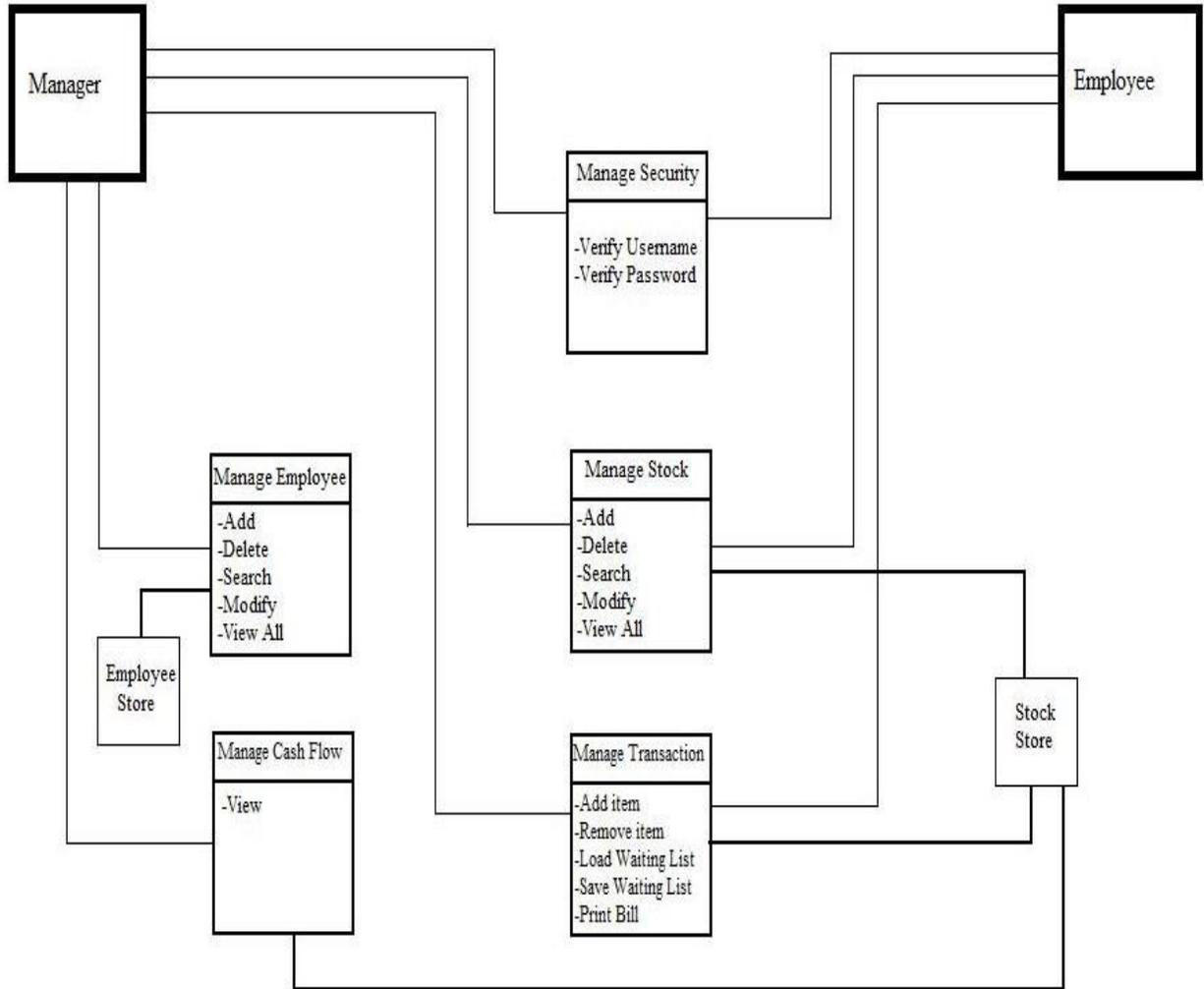


Figure 1: Software Requirements

## 2.3 Design

The design is a demonstration of our methodology and the form under which the system functionalities are grouped and presented to the user. This stage emphasizes the main entities in the system and how they interact with each other.

Figure 2 highlights the system entities and their interaction.



**Figure 2: Software Design**

## 2.4 Testing

To verify that ShopSYS is fully functional and meets the goals set during the requirements elicitation phase. The next chapter in the development of ShopSYS was to trial the system with different, thorough, and exhaustive test cases. The testing phase is the last chance to make modification, fix bugs, and confirm that the application software meets its requirements before presenting it. Test cases are created separately and the developer of the software has to anticipate the result before trying the case on the system. If the anticipated result and the real one do not match, further adjustments to the system are required.

The following table represents some test cases for manage security and cash flow functions. Test cases for manage employee, stock, and transaction are not included in the consideration of space.

**Table 2: Test Cases for Manage Security and Cash Flow**

Process	Test Case	Result
Manage Security	No username or no password entered (ex: password text field left empty)	Access denied and message informing about wrong input
	Username and password equal to (manager, manager)	Access to the manager home frame
	Username and password equal to (employee, employee)	Access to the employee home frame
	Trying one of the two combinations above with different cases. (ex: mANaGer, mAnageR )	Access to the correspondent frame(manager or employee)
	For any combination different than the ones above: (ex: manager – employee; employee-aaa )	Access denied and message informing about wrong input
Manage Cash Flow	Launching the system in a new month (ex: last time the program was launched was in march and the current month is April)	New month revenues and expenses sheet automatically added
	Performing a transaction on March 15	Revenues and benefits in the March cash flow sheet updated
	Performing a <u>transaction</u> in a new month (ex: last time a bill was printed launched was in march and the current month is April)	New month sheet automatically added and updated using values from transaction

### 3. RESULT

The testing stage proved that all developed functionalities are working and that our final product could meet its initial requirements and objectives. The system was tried on several computers with different operating systems and specifications and the result was perfect.

To make the software more professional and user friendly, we created a SETUP tool which will install ShopSYS by executing it. As stated before, to successfully install ShopSYS, users will need a license key generated by our own key generator. Figure 3 is a snapshot of the license key generator. SETUP tool also includes a ReadME text file which offers the user some hints on how to use ShopSYS. The tool also adds to the hard disk an uninstall executable and a folder where snapshots of all customer bills are kept. Finally, it adds a shortcut of ShopSYS to the desktop of the user.



**Figure 3: Key Generator**

The first thing that users meet when they start ShopSYS is the main frame which allows them to log in. The Figure 4 represents the main frame. In addition to logging in, the main frame allows the manager to create accounts for new employees so that they can start accessing ShopSYS. It also offers users some help in case they forget their logging in information or can't get around ShopSYS as smoothly as they want to. When users access their accounts, they move to a new frame based on their access level. Figure 5 represents the home of a manager. From this frame, managers and employees can run their store efficiently using different functionalities. When done, they can simply log out and go back to main frame or close the program directly.

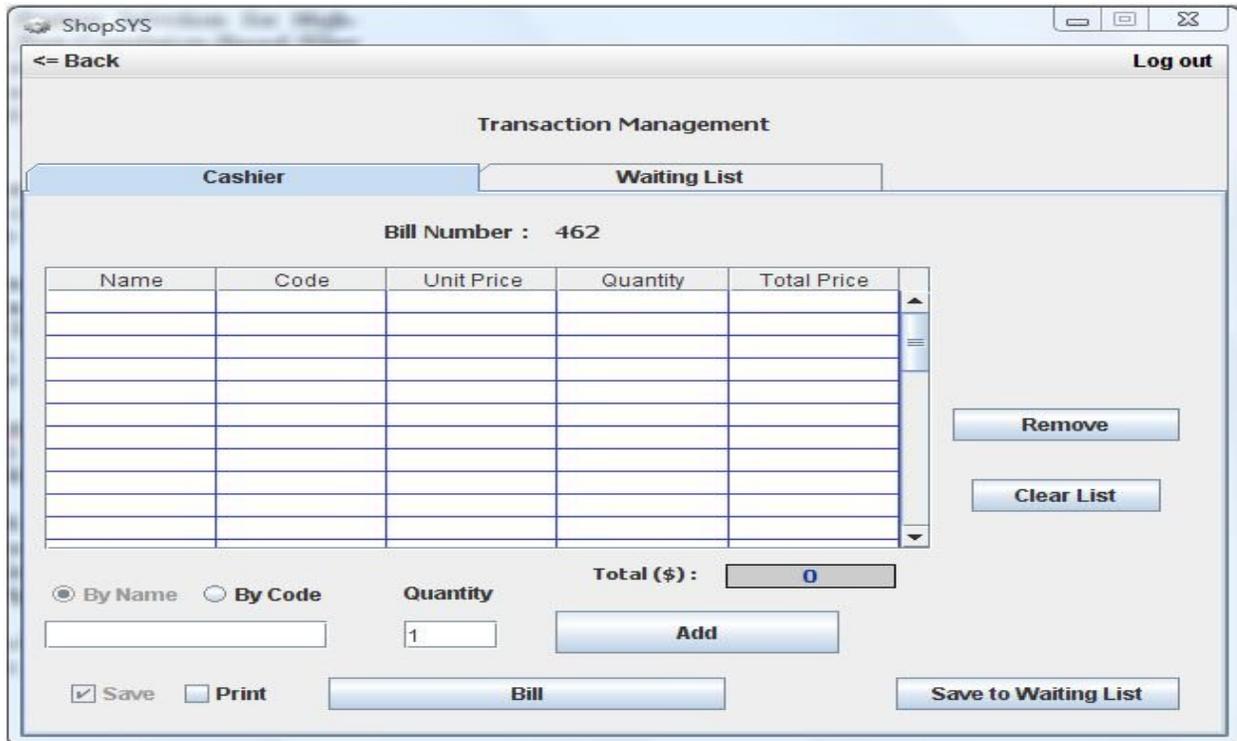
As we said before, the main functionality that will be used is the manage transaction utility. It is the main reason why shop owners need an automated system. Figure 6 shows our transaction management frame. Since the largest part of the job will be managing transaction, we decided to put all function on one form to avoid going back and forth. Users will be able to add items by name or code and they can specify the quantity to avoid repetitive items. They can print bills or not depending on their choice but a copy of the bill is always saved on their hard disk. They can also move the bill directly to the waiting list and process it in the future.



Figure 4: Main Frame



Figure 5: Manager's Home Frame



**Figure 6: Transaction Frame**

#### 4. CONCLUSION AND FUTURE WORKS

We designed, implemented and tested ShopSYS, a software application that facilitates small shop businesses to manage their daily operation. It allows employees to manage stock and create customer bills. It also permits for managers to keep record of their employees and keep track of the cash flow in their store. The application was created as a project for software engineering course during the spring 2008 semester. Using theories and models learned during the same course, ShopSYS was developed in a systematic, disciplined, and proficient approach. All functions of the software are working properly at the end of the project and we received positive feedback from the demo at the end of the semester.

As any other application, ShopSYS is open for further changes and additions in the future. One major update is to allow different computers through the same network to access the same stock database. All cashier stations in the same business should be able to access and update the same database since they share the same stock in real life. To make this addition much easier, it is better to change the data handling approach from normal text files to a professional database management system such as Oracle or Microsoft SQL Server. Text files are trustworthy and sufficient to serve a small business with a little number of employees and items in its store. However, they are very hard to handle in a network and less secure than a professional database system. Moreover, a good addition to the system would be to add a digital signature to all customer bills. The digital signature will reflect which employee or manager approved that bill so if a mistake is done, it will be much easier to trace back which user of the system did it. Last but not least, most products in real life have numerical codes which are used to reflect the name and price of the item. We could enhance the transaction function with the ability to add items to bill using an infrared device that reads items' codes.

In such dire economical times, small shop owners and small businesses in general are in a greater need for simple and economical software that could help them to grow their business. ShopSYS as a project gives a great illustration of how to create a system that will suit the needs of a small business in a short period of time and with limited resources. It can be used as a model for professional software development in the future.

## 5. REFERENCES

- [1] I. Sommerville, Software Engineering, Addison Wesley Publishing Company, 2006.
- [2] B. Eckel, Thinking in Java, Prentice Hall PTR, 2006.
- [3] Java Tutorial by Sun Microsystems. <http://java.sun.com/docs/books/tutorial>
- [4] S. Berouel and A. Adams. ShopSYS Project Report, Software Engineering Course at the University of the District of Columbia, Washington D.C, April 2008.