BUSINESS MANAGEMENT SYSTEM

A comprehensive management system for small business.

Name: Bingyao Tian

Student ID: 1605020

Degree: MSc Computer Science

Supervisor: Dr Ian Kenny

School of Computer Science, University of Birmingham

11/01/2016

Table of Contents

Abstract

This project is about building a software for small business owners which helps them deal with their daily activities such as sales entry, managing human resources from the traditional method. The project analysis both functional and non-function requirements from some small business owners. This software-based project follows the waterfall model and software is Java based working with MySQL database. In the functional part, it allows users to login, make transactions entry, manage asset and people and view financial statements. In the non-functional part, the software is easy to use with advantages of performance, safety and reliability compare to traditional hand writing and book keeping.

ACKNOWLEDGEMENTS

I would like to acknowledge the help given to me by my supervisor, Dr. Ian Kenny and my inspector, Dr. Kashif Rajpoot, thanks for every help during this project period.

I would also like to acknowledge those who filled my questionnaire. That was invaluable information to help determine the issues this project trying to solve, and requirements of the software.

Finally, my family and my friends, who have been very supportive to me, not only throughout the duration of the project but throughout the whole year.

Introduction

The aim of the work described in this report was to provide a software with which small business owners can use for daily business management activities instead of the traditional booking keeping method to improve performance and information security.

The issue was identified by using use case and interviewing 16 individuals including wholesale business owners and manufactory business owners in Linyi, China. The city which has the second largest wholesale market in China, according to Linyi Trade City Administrative Commission, there are 128 specialized wholesale markets with 50,000 merchants engaged the dealing of more than 60,000 kinds of products covering 27 categories. Around the trade city over 200 logistics enterprises waved a huge logistics system which can directly connect to over 1,800 cities above county level and all ports around the country.

As such a business city only 15.4% of the small businesses are currently using computer technology to do office work according to the results collected from questionnaire (Appendix. 1). The questionnaire form indicates the issues with the traditional method of book keeping mostly are easy to lost, difficult for extracting information and analysis, inefficient.

To solve this problem, the project is to provide a software which integrate accounting and management functions, and relatively traditional office model it improves security, the system will help users to do work more intelligent. Consider the target users are mostly small businesses, the system should be relatively simple and less cost.

The software development process follows the waterfall model, which is a sequential design process, flowing steadily downwards (like a waterfall) through the phases of requirement analysis, design, implementation verification and maintenance.

Figure 1 shows the steps of waterfall model process.

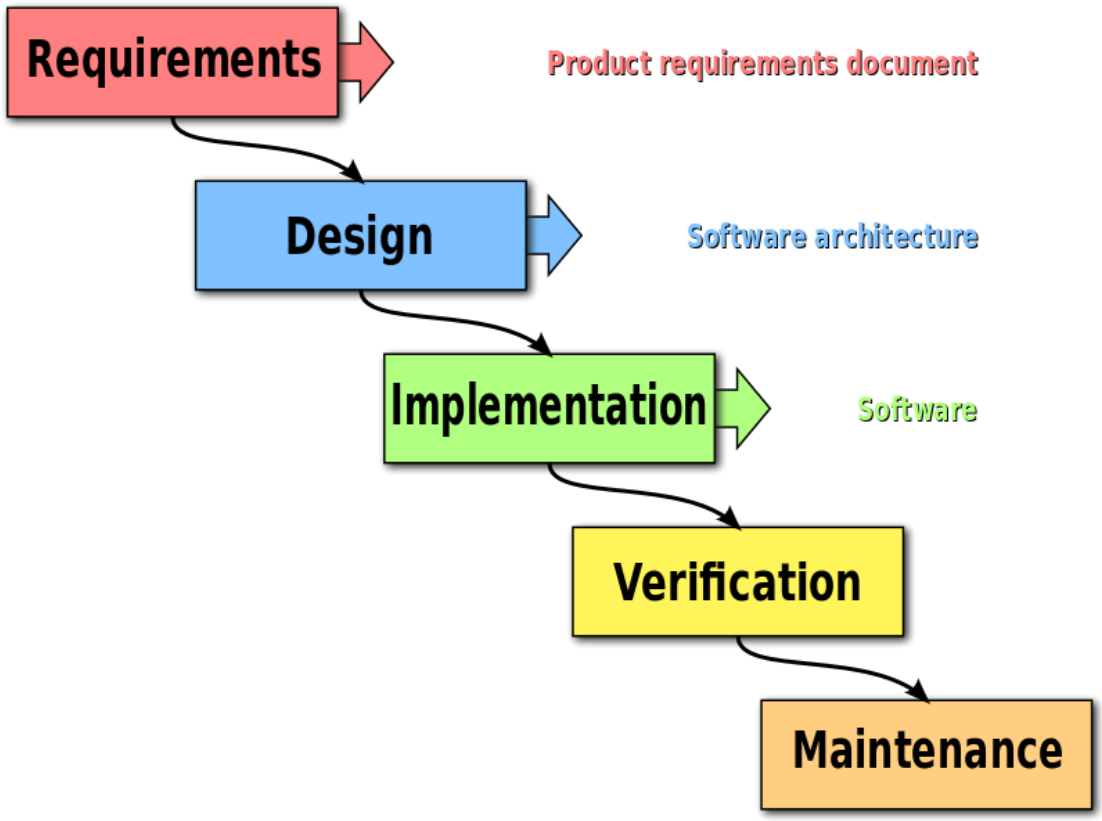


Figure 1. Waterfall Model

The waterfall model offers numerous advantages for software developers. First, the staged development cycle enforces discipline: every phase has a defined start and end point, and progress can be conclusively identified (through the use of milestones) by both vendor and client. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage, or of customer expectations not being met. Getting the requirements and design out of the way first also improves quality; it's much easier to catch and correct possible flaws at the design stage than at the testing stage, after all the components have been integrated and tracking down specific errors is more complex. Finally, because the first two phases end in the production of a formal specification, the waterfall model can aid efficient knowledge transfer when team members are dispersed in different locations. (Melonfire, 2006)

Structure of the Report

This following sections of this report are corresponding to the first 3 phases of waterfall model, in the later of this report, it contents the “Walkthrough” to introduce how to use the software. Then testing is to detect the bugs and errors of the software. After that, evaluation is to describe the results and how well the software does in use. Discussion is to summaries the achievements and also the deficiencies of the project. Conclusion is to describe how the project solve the problem addressed in the first place. The References and Appendices will be attached at the end of this report.

Analysis and Specification

This first step is requirement analysis, which is the most important part, because it involves gathering information about what the user needs and defining, in the clearest possible terms, the problem that the product is expected to solve. Analysis includes understanding the user's business context and constraints, the functions the software must perform, the performance levels it must adhere to, and the external systems it must be compatible with. Techniques used to obtain this understanding include potential user interviews, use cases. The results of the analysis are typically captured in a formal requirements specification, which serves as input to the next step. (Melonfire, 2006)

The question form made for collecting information about the requirement from potential users is provided by Tencent, one of the largest and most used internet service portal in China. The question form is sent to Mr. Xianghua Tian, who is the owner of Linyi Huaneng building material Co., Ltd. Then Mr. Tian send to his local business friends, and so on. The result indicates that most of the interviewers are manufactory and wholesale business owners, 84.6% of them are currently using the traditional way of doing their office work (paper and pen), inefficient, easy to lose and not environmental friendly are the most concerned issues, most of them are optimistic about the future of this software. For the functional requirements, transaction records and people management (include customers , employees and suppliers) are selected at a large proportion. The most concerned non-functional requirements by the interviewers are efficiency, usability, portability and security. (Appendix 1)

The other technique used for identifying the requirement is use case diagram, which is a representation of the interactions between users and the system. A use case diagram can identify the different types of users and different use cases of a system.

Specifically, a use case diagram basically has actors, use cases and the system. Actors are the users of the system, use cases are the tasks actors undertake with the system.

The use case diagram for this business management system is basically having 3 actors: business owner (root user), employee (non-root user) and IT support. Employee can log in, enter sales record, produce invoice, enter purchases, manage assets, manage inventories, enter expenses records, enter liabilities, view financial statements, view and edit customer, supplier and employee information. Business owner has one more use case compare to employee, business owner can add or update system user (non-root user). IT support do the maintenance work for the system. Figure 2 shows the use case diagram with user cases for the 3 actors in the software.

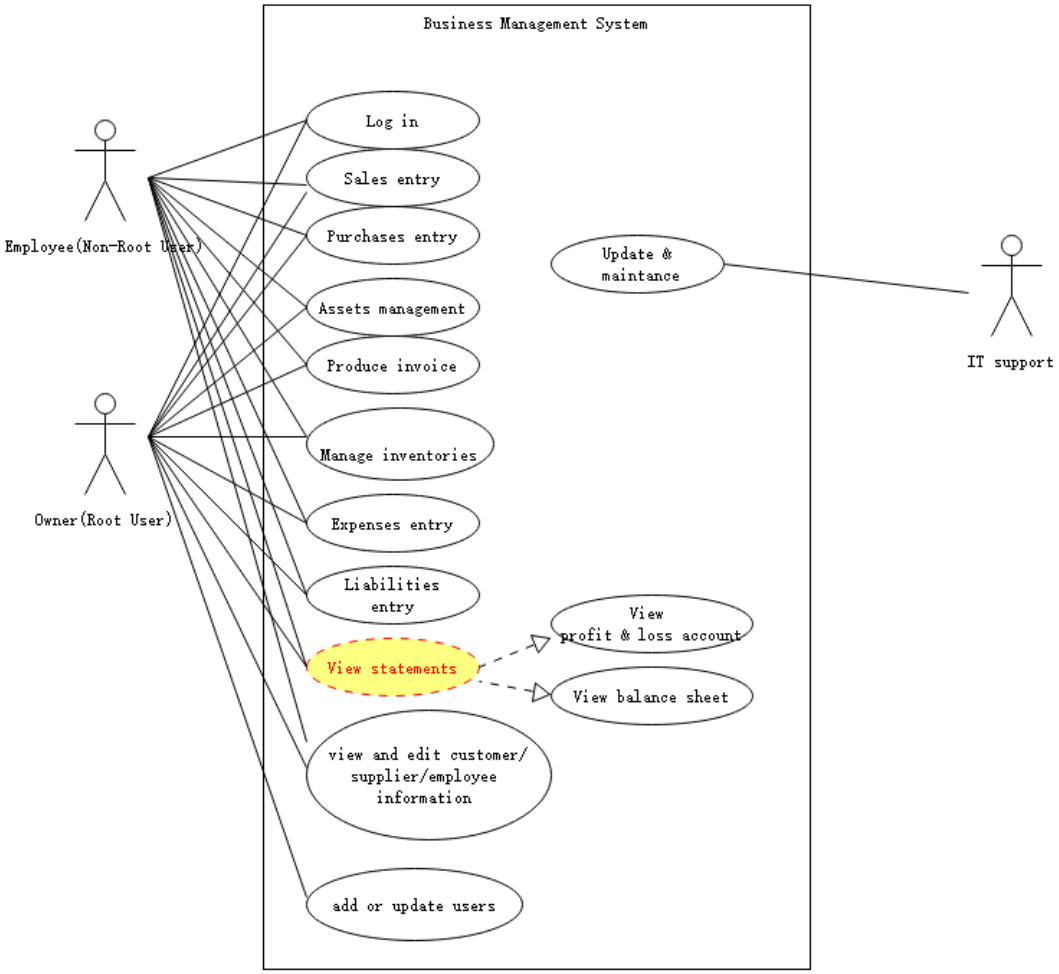


Figure 2. Use case diagram

User Requirements

Functional Requirements

Essential

1. The program must allow user to log in with unique username and password.
2. The program must allow user to entry sales, purchases, liabilities and expenses.
3. The program must allow user to make changes to assets.
4. The program must allow user to manage inventories.
5. The program must allow user to view financial statements.
6. The program must allow user to view and edit customer/supplier/employee information.
7. The program must allow Root user to add or update users.

Desirable

1. The program could have a homepage to display business news from selected news provider.
2. The program could have a reminder tool on homepage to remind user some important business activities.

Non-functional Requirements

1. **Safety and reliability**
   The system should run normally without crashing, the data should be stored safe.

   Hand writing is the traditionally way for managing business, it's easy but there are many security issues. Paper is easily damaged due to many situations: missing, fire and wet, etc. More importantly, it is unlocked, everyone will see the information if they have the paper. So with those concerns, small business owners need a system to replace the traditionally way with better safety guarantee.

2. **Performance**
   The system should have better performance than the traditional method.

Efficiency is a key factor for transmitting from traditional way to IT style. With a computer software, many things can change dramatically, for example, system may automatically issue an invoice as soon as a sale has been made, system may automatically calculate VAT, etc.

3. **Usability**
   The system should be useable for normal people, easy to understand, only few people in the world are genius.

4. **Cost**
   The system should not cost too much to implement.

   With the traditionally way it won't cost much, so if small businesses use computer system to replace it, the computer system should do even better.

5. **Portability**
   The system must operate in the popular computer operating systems, such as Microsoft Windows, Apple Mac OS X and the popular versions of Linux.

6. **Ethical**
   The system must have no unpleasant bundle and the system must not leak any users personal information.

Design

This step consists of "defining the hardware and software architecture, components, modules, interfaces, and data...to satisfy specified requirements". It involves defining the hardware and software architecture, specifying performance and security parameters, designing data storage containers and constraints, choosing the IDE and programming language. This is also the stage at which user interface design is addressed, including issues relating to navigation and accessibility. The output of this stage is one or more design specifications, which are used in the next stage of implementation. (Melonfire, 2006)

The general idea of this software is that user can login at the login window with unique username and password, if both username and password are correct, then go to the main window, user can do the activities showed Figure 2.

The system architecture is basically *Model-View- Controller (MVC)* pattern, which separates the modeling of the domain, the presentation, and the actions based on user input into three separate classes (Burbeck, 1992). The *Model-View- Controller (MVC)* pattern makes later modifications and extensions to the program simple and makes it possible to reuse some parts of the program.

- Model. The model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).
- View. The view manages the display of information.
- Controller. The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.

Figure 3 depicts the structural relationship between the three objects.

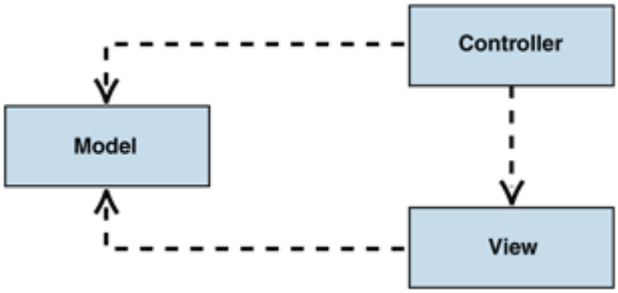


Figure 3: MVC class structure (Burbeck, 1992)

Figure 4 shows the behavior of the MVC pattern classes.

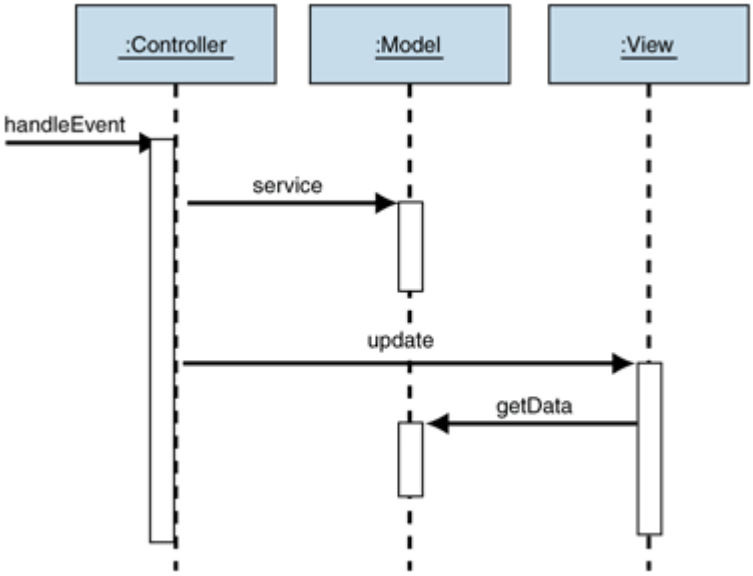


Figure 4: Behavior of the passive model (Burbeck, 1992)

Basically, the classes used can be divided into four categories:

- **Database class.** Connect to database, store methods to manage the data, execute SQL queries.
- **Client class.** Communicate between GUI and Database.
- **Component classes.** Classes maintain data about those entities.
- **Graphical User Interface classes.** This category includes the main user interface classes, including the class responsible for the login in window and main window of the program.

Programing language

Consider the cross-platform requirement, Java is used for this program. Java is one of the most popular programming language with a reported 9 million developers in 2013. (Beneke and Wieldt, 2013) One of the most significant advantages of Java is its platform independency, Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. The business management system can run on any machine as long as there is JVM installed on the machine.

Database

The database is used to store the data for the business management system, including users' username and password, sales records, employees' information records, etc. MySQL is an open-source relational database management system. Compared with other large databases such as Oracle, IBM DB2, MS SQL, etc., MySQL has its own shortcomings, such as small size, limited functionality, but for the normal individual user and small and medium enterprises, MySQL provides the functionality is more than enough, and because MySQL is open source software, so you can greatly reduce the total cost of ownership. MySQL is choosing for the database for this program.

Database structure:

The name of the database used for the business management system is "project", Figure 5 shows the tables inside the database project, the detailed database structure is attached in appendix 2.

Assets	Customers	Employees	Expenses	Inventories	Liabilities	Notes
Payables	Purchases	Receivables	Sales	Suppliers	Users	

Figure 5: The tables inside the database "project".

User Interface Design

When start the system, there is a login window for user to enter username and password. Figure 6 shows the basic idea for the login window.

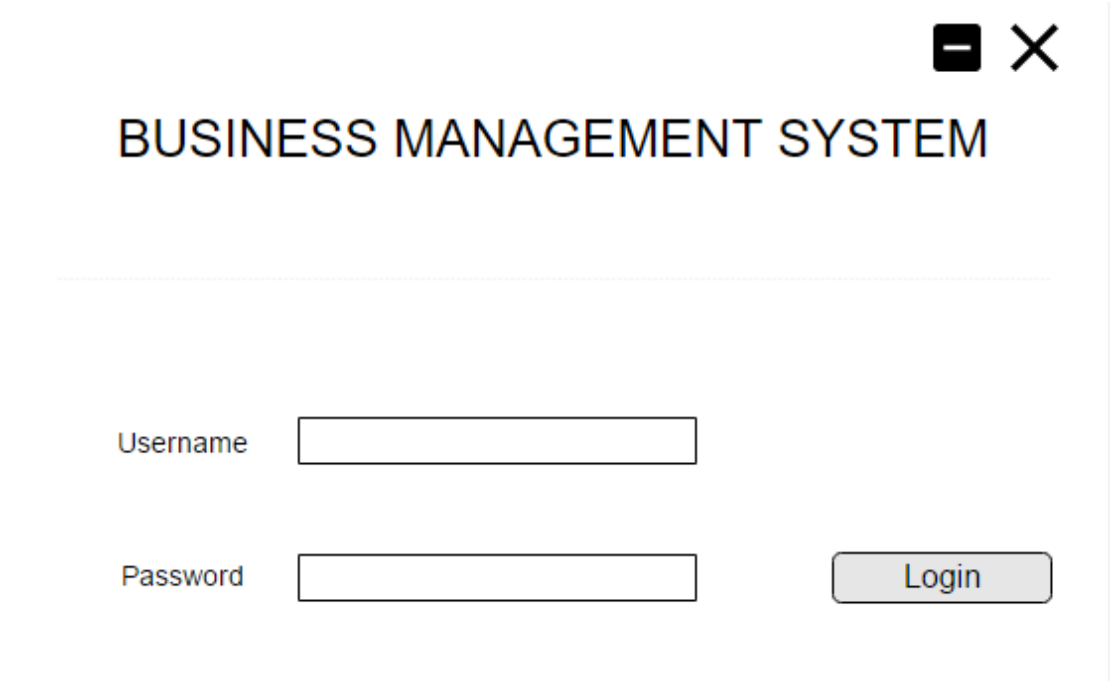


Figure 6: login in window of business management system

If the login check pass, the login window will be closed and open a new window automatically, which is the main window of the application.

The main window consists of multiple tabs includes:

- **Homepage**. Overview of business statues.
- **Sales**. Add and view sales records.
- **Purchases**. Add and view purchases records.
- **Inventories**. Add and view inventories records.
- **Assets**. Add, update and view assets records.
- **Expenses**. Add and view expenses records.
- **Customers**. Add and view customers' records.
- **Suppliers**. Add and view suppliers' records.
- **Employees**. Add, update and view employees' records.
- **Liabilities**. Add, update and view liabilities records.

- **Financial Reports**. View financial statements, including income statement and balance sheet.
- **Settings**. For root user only, to add, update, delete and view the users of the system.

Figure 6 shows the Homepage of the main window for the business management system. The top side are 4 textPane showing current date and time, current monthly sales, current receivables and payables of the business respectively. The left side shows the business news subscripted by the user. The right side is a reminder tool.

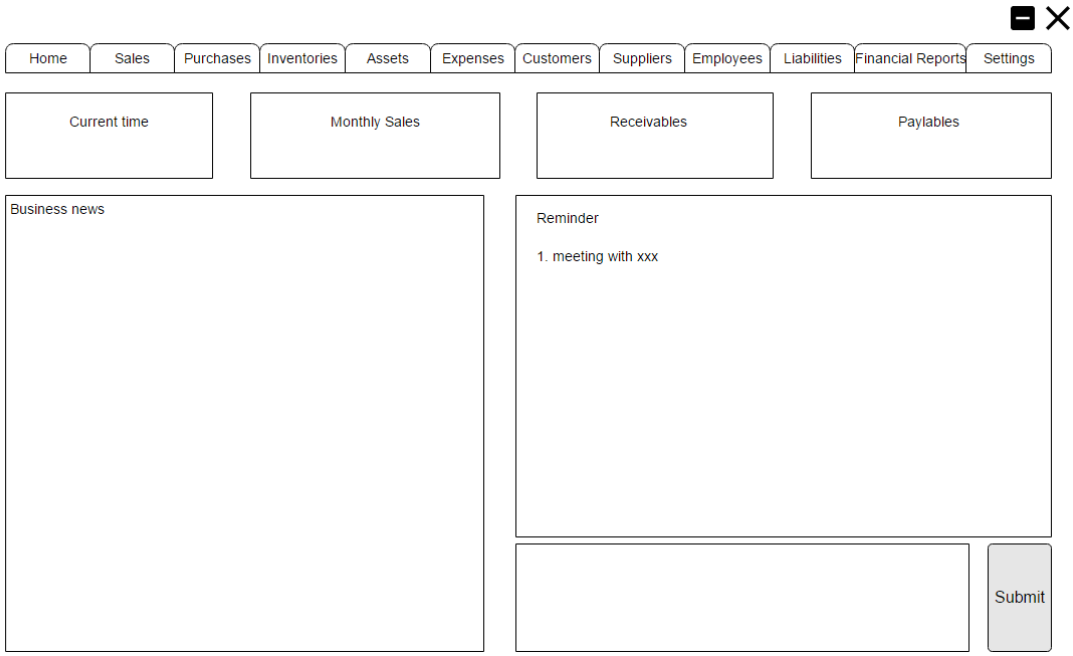


Figure 6: UI design for the homepage for the business management system.

Figure 7 shows the idea of sales tab of the main window for the business management system. On the left side, there is a textArea showing the information required by the user. On the right side, user can add sales records and search sales records. The rest tabs are more or less like the sales tab.

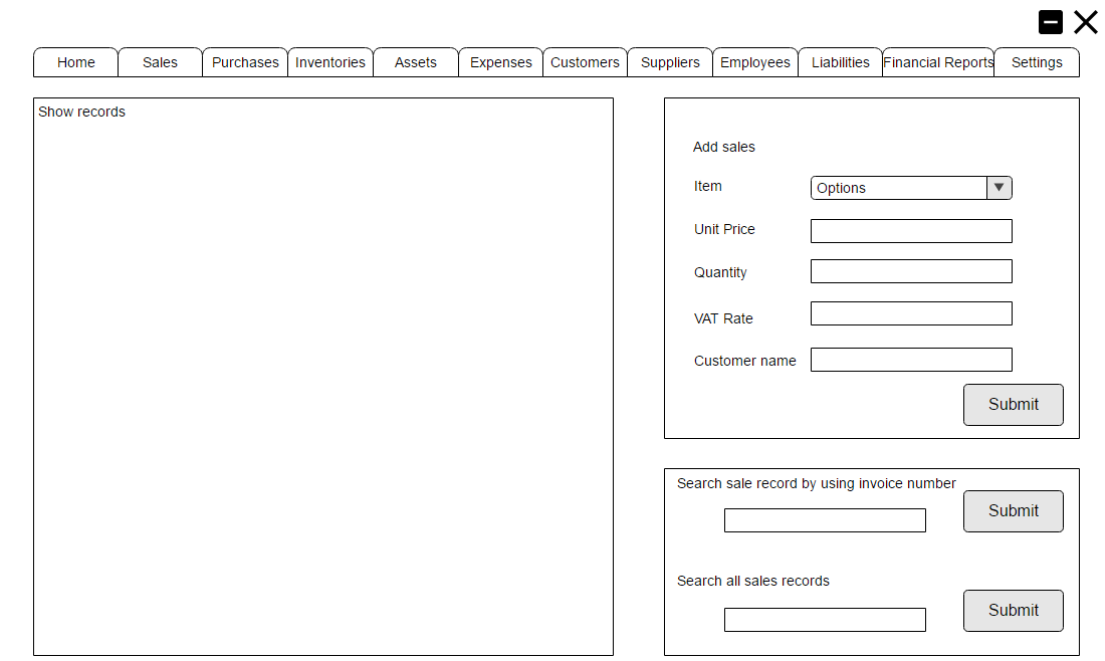


Figure 7: UI design for the sales tab for the business management system.

Implementation

This step consists of the actual constructing the product as per the design specification(s) developed in the previous step. The detailed class diagram of the business management system is attached in Appendix 3. The output of this step is basically the final product.

Component package

This package includes all the component classes for the program, such as Assets, Customer and Employee. Every component class has its own field variables, constructer, getters, setters and a toString method. The constructer, getters and setters are generated by Eclipse. For example, Figure 8 shows the Assets class has field variables name, date, quantity, unitValue and totalValue with types String, Date, int, double and double respectively. Then the Assets class has a constructor public Assets with parameters name, data, quantity, unitValue and totalValue. Finally, it has getters, setters and a toString method print the information of an asset.

```java
 * @author Bingyao Tian
 * @version 2016-11-11
 */
public class Assets {
    private String name;
    private Date date;
    private int quantity;
    private double unitValue;
    private double totalValue = quantity*unitValue;

    /**
     * Constructor of an asset.
     * @param name
     * @param date
     * @param quantity
     * @param unitValue
     * @param totalValue
     */
    public Assets(String name, Date date, int quantity, double unitValue, doubl
        this.name = name;
        this.date = date;
        this.quantity = quantity;
        this.unitValue = unitValue;
        this.totalValue = totalValue;
    }

    /**
     * getter
     * @return name
     */
    public String getName() {
        return name;
    }

    /**
     * setter
     * @param name
     */
    public void setName(String name) {
        this.name = name;
    }
    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }
    public int getQuantity() {
        return quantity;
    }
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public double getUnitValue() {
        return unitValue;
    }
    public void setUnitValue(double unitValue) {
        this.unitValue = unitValue;
    }

    public double getTotalValue() {
        return totalValue;
    }

    public void setTotalValue(double totalValue) {
        this.totalValue = totalValue;
    }

    /**
     * toString method to print the information of an asset.
     */
    @Override
    public String toString() {
        return "Asset name: "+name +"\n"+ "Quantity: " + quantity + "\n"+"Unit

    }
```

Figure 8: Assets class for maintaining the data of assets.

Database class

The Database class plays the *Model* roll in the *Model-View-Controller* partten, it has a method to connect to MySQL database via Java Database Connectivity (JDBC). JDBC is an API for Java defines how a client may access a database. Then the database class includes methods to execute specific SQL queries. For example, Figure 9 shows a method for checking whether or not the username and password entered are correct.

```
public boolean loginCheck(String username, String password){
    conn = getConnection();
    String query = "SELECT username, password FROM user";
    String dbUsername, dbPassword;
    boolean login = false;

    try {
        statement = conn.createStatement();
        statement.executeQuery(query);
        ResultSet rs = statement.getResultSet();

        while(rs.next())
        {
            dbUsername = rs.getString("username");
            dbPassword = rs.getString("password");

            if(dbUsername.equals(username) && dbPassword.equals(password))
            {
                System.out.println("OK");
                login = true;
            }
        }
        conn.close();
        }
    catch (SQLException e) {
        e.printStackTrace();
    }
    return login;
}
```

Figure 9: A method for checking if the username and password are correct.

Specifically, the method passes two parameters, username and password which are entered by user, and compare with the username and password extracted from database, if both username and password entered equals the username and password

stored in the database, the method will return true, then let the user login, otherwise not.

Client class

The Client class plays the controller roll in the *Model-View-Controller* partten, it is the class that interprets the mouse and keyboard inputs from the user, and work together with the methods from Database class to process the information then change the view appropriately. For example, Figure 10 shows the LoginListener class that intercepts the user's inputs for username and password, and call the loginCheck method from Database class to verify the login. If the loginCheck method returns true, the login window close and open the main window of the business management system. Otherwise it display an error message.

```
class LoginListener implements ActionListener{
    public void actionPerformed(ActionEvent arg0) {
        String username, password = null;
        try{
            username = loginView.getUsername();
            password = loginView.getPassword();

            if(Model.loginCheck(username, password)==true){
                System.out.println("success");
                EventQueue.invokeLater(new Runnable() {
                    @Override
                    public void run() {
                        try {
                            Homepage frame = new Homepage();// open Homepage
                            frame.setVisible(true);
                            loginView.dispose();
                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                });
            }else {
                loginView.displayErrorMessage("Username or password wrong!");
            }
        } catch(Exception e){
            loginView.displayErrorMessage("Wrong");
        }
    }
}
```

Figure 10: LoginListener class.

Graphical user interface classes

There are 3 classes under package GUI to play the role of *View* in the *Model-View-Controller* partten. The toolkits for building the GUI for the business management system is mainly Java Swing toolkit with some parts of Java Abstract Windowing Toolkit (AWT), such as ActionListener and GridLayout which are only available from Java AWT. The biggest difference between them is that AWT is a Java interface to native system GUI code present in your OS, it will not work the same on every other system. Swing is a more-or-less pure-Java GUI. It uses AWT to create an operating system window and then paints pictures of buttons, labels, text, checkboxes, etc., into that window and responds to all mouse-clicks, key entries, etc., deciding for itself what to do instead of letting the operating system handle it. Thus, Swing is 100% portable and is the same across platforms.

Login class

Login class creates a window for user to login, it contents textFields to receive the input from user and button to listen to the action.

Homepage class

Homepage class is the main window of the software, it basically has different tabs every tab has its own functions, this will be seen in the next section “User Interface Walkthrough”.

ProjectStart class

The ProjectStart class is for start the program, it has main method to initializes a new Login window.

User Interface Walkthrough

This section provides a number of screen shots for the software, and the explanations are given as to the related functions.

Login window

After start the software, the user is presented with a login windows which requires the user to type in the username and password.

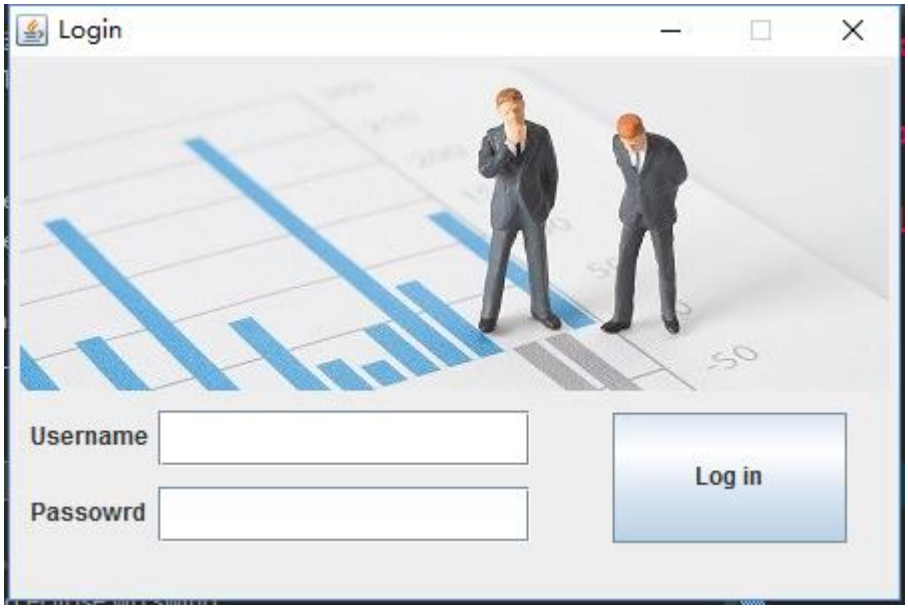


Figure 11: The screen shot of login window.

If the username or password not correctly entered, it will display error message.

Main window

Overview tab

Once the login check passes, the main window of the business management system is presented.

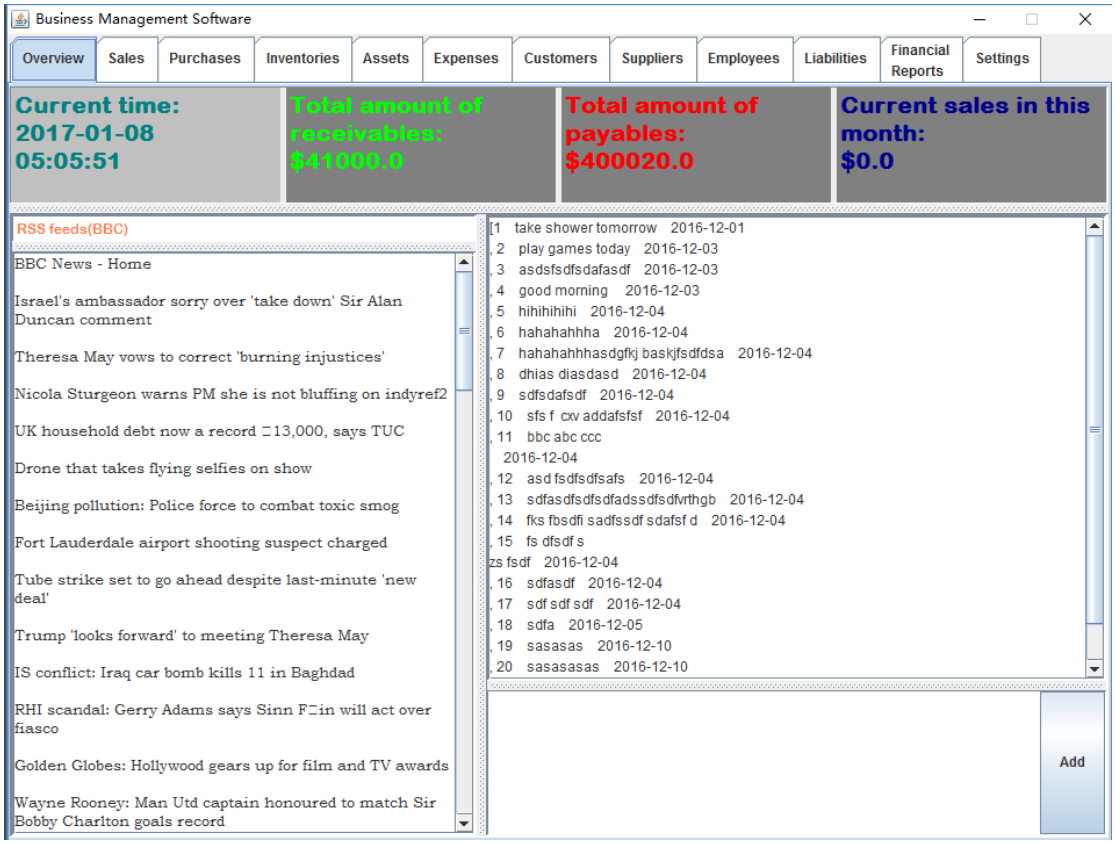


Figure 12: The screen shot of the Overview of the software.

On the top, there are the navigations of each tab. There are 4 grey textPanel showing the current time, receivables, payables and monthly sales. Down left is the RSS news feeds. Right side is the reminder tool, user can enter message inside the bottom textArea, click “Add” button the message will be displayed.

Sales tab

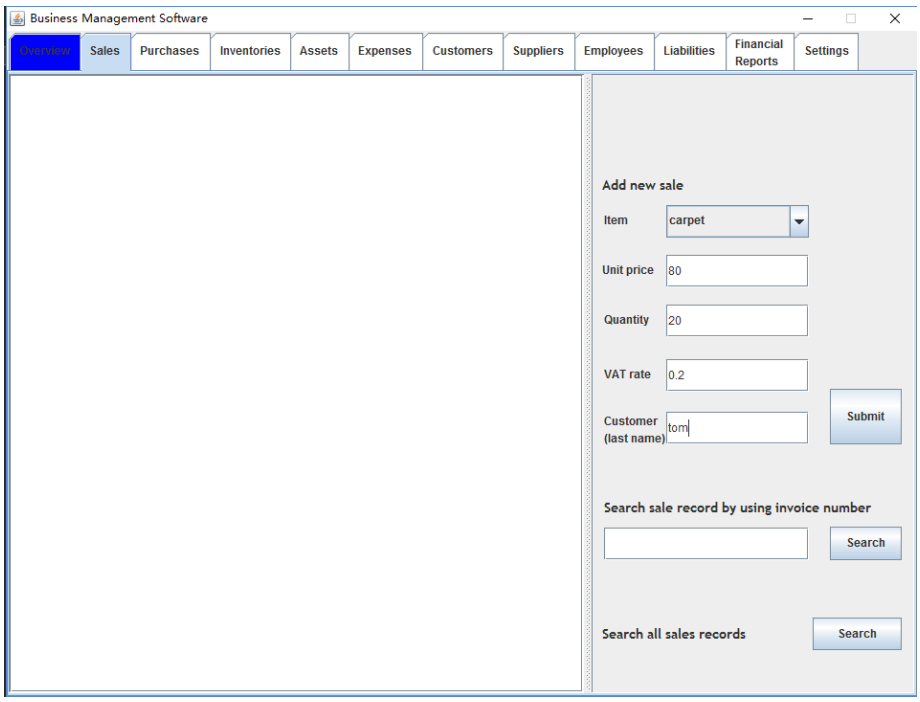


Figure 13: The Screen shot of the Sales tab of the software.

The Sales tab is used for user to make sales and view sales records. Figure 13 shows a case, when a order requested by tom about selling 20 carpet at 80 each with additional 20% VAT. The drop-down list lists the inventories for selling. If the stock is not enough for the order, it will show a warning message.

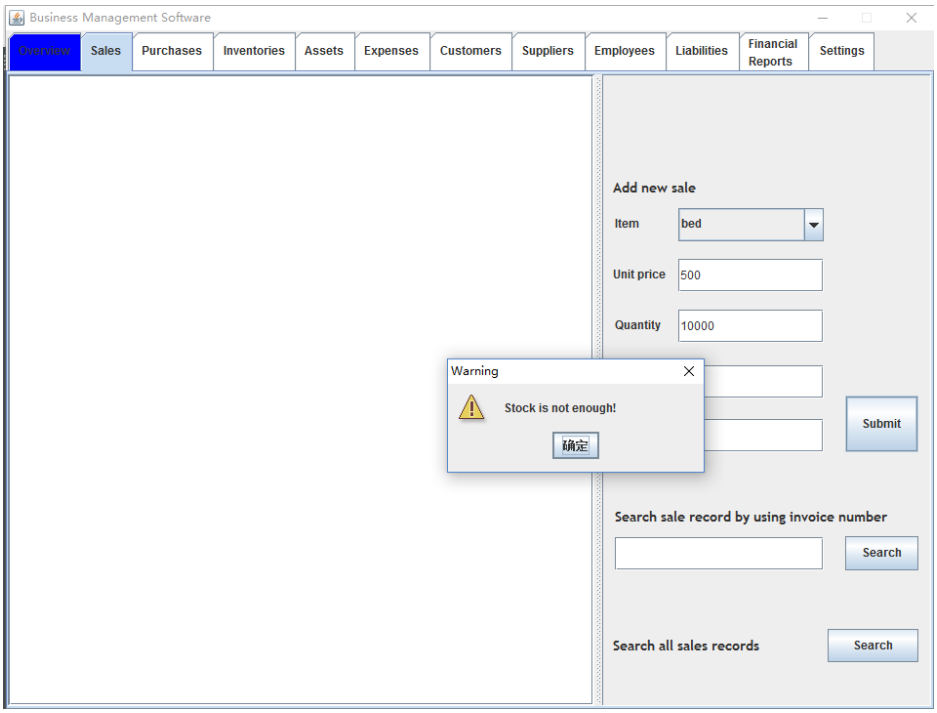


Figure 14: The screen shot of the Warning message for not enough stock.

The invoice number is automatically generated by the system once an order is made,start at 100001, user can use the invoice number to see the details for that sale. Additionally, the “Search all records” allows user to view all the sales records.

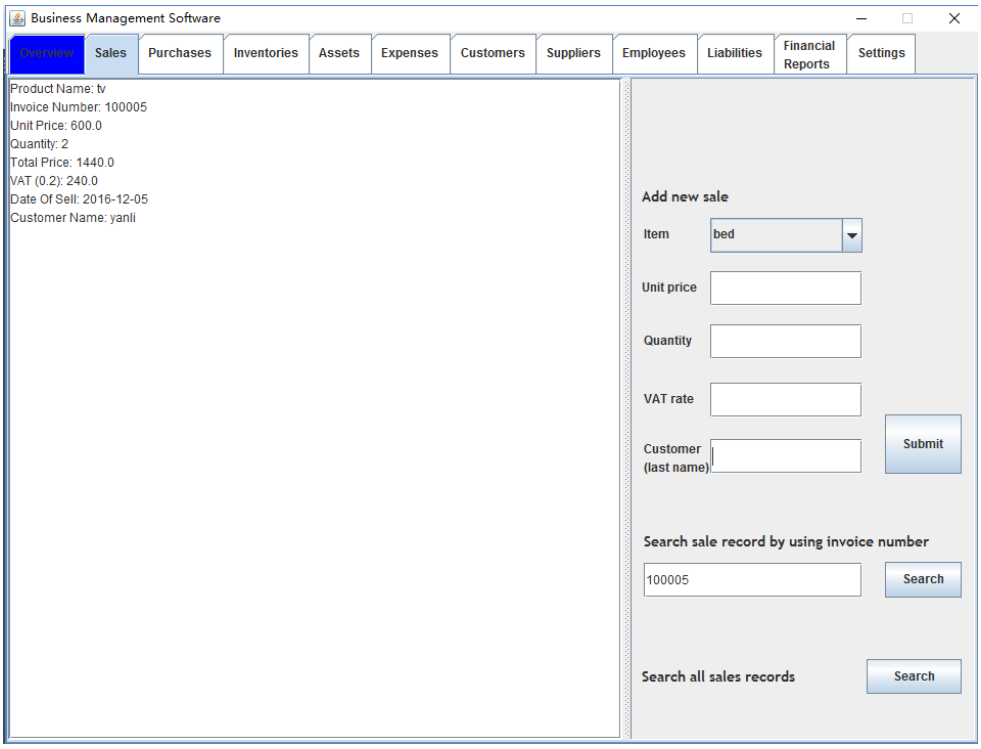


Figure 15: The screen shot of searching a specific sales record by using invoice number.

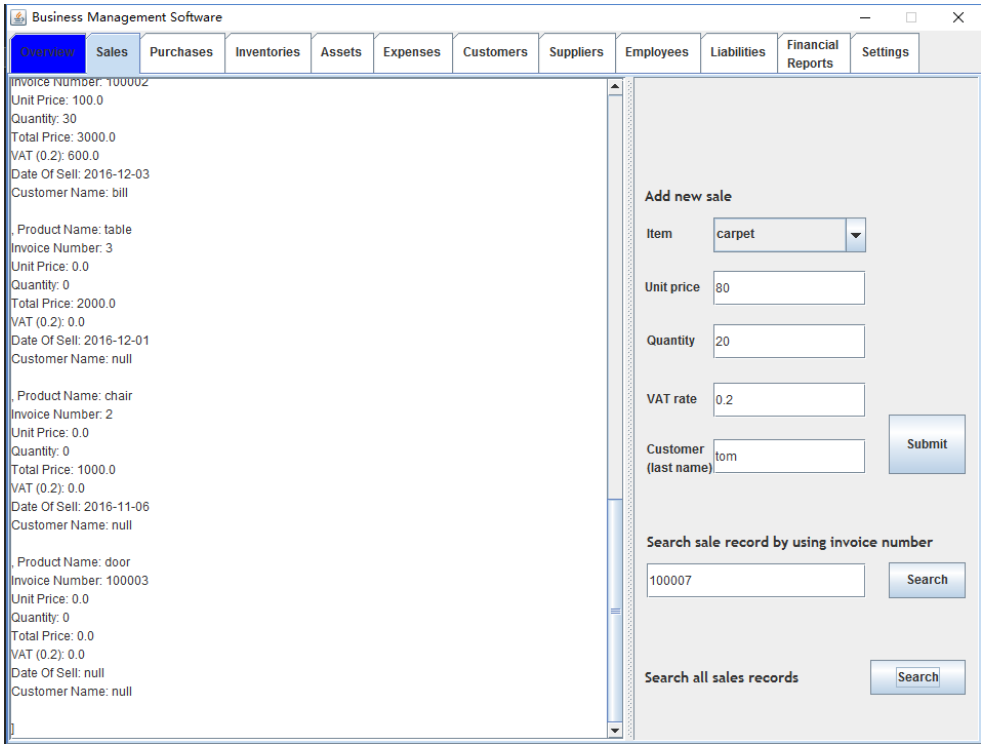


Figure 16: The screen shot of viewing all sales records.

Financial Report tab

"Financial Reports" tab allows user to view the income statement and balance sheet by entering the financial year which user want to view. The system uses the first calendar day (1st of January) of every year as the start of the financial year, and uses the last calendar day (31th of December) of every year as the end of the financial year.

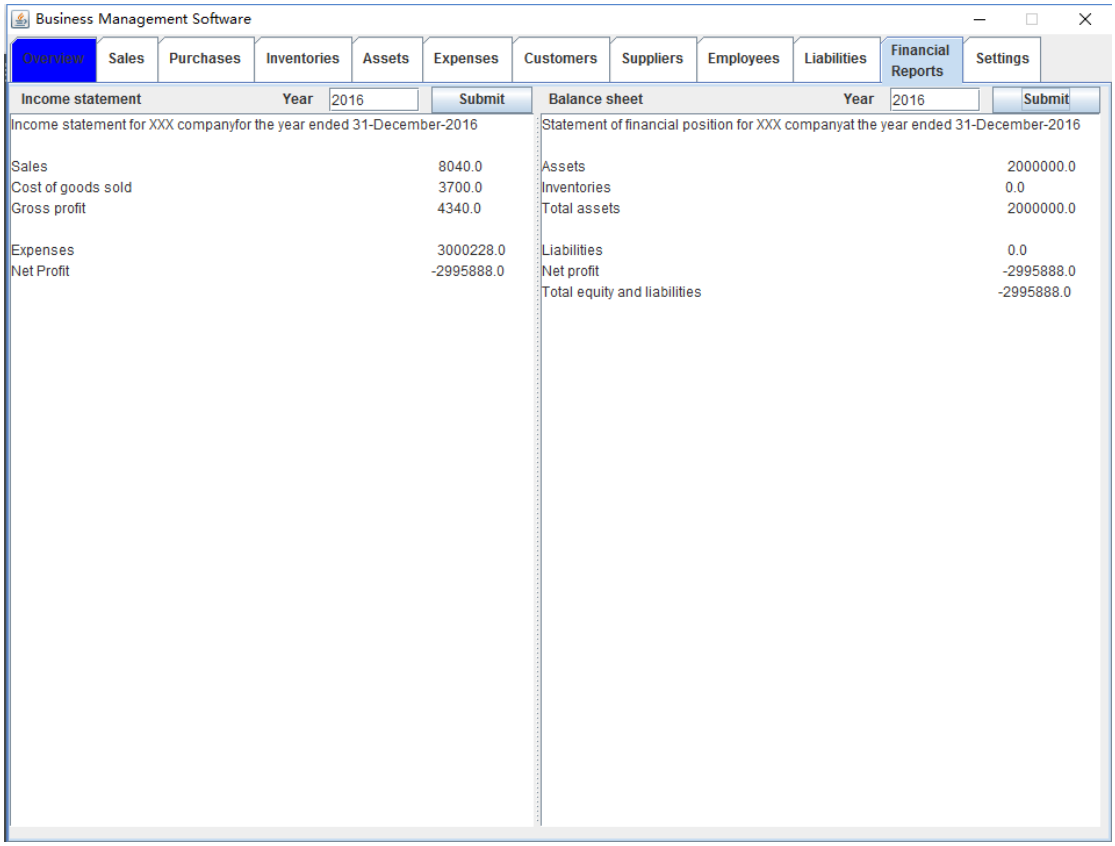


Figure 17: Screen shot of the "Financial Reports" tab showing the income statement and balance sheet for the year 2016.

Income statement

The income statement is the statement that shows the financial performance of the business during the financial year. The sales in the income statement is the total sales during that year. Cost of goods sold is the total cost of the products sold during that year, which is calculated by using the formula:

Cost of goods sold = Opening Inventories + Purchases – Closing stock

The gross profit is calculated by using the formula:

Gross profit = Sales – Cost of goods sold

The net profit is calculated by using the formula:

Net profit = Gross profit - Expenses

Balance sheet

The balance sheet also called the statement of financial position, which is a financial statement that summarizes a business's assets, liabilities and shareholders' equity at a specific time.

The total assets of the business is calculated by using the formula:

Total assets = Assets + Inventories

Total assets = Total liabilities + Net profit + shareholders' equity

Other tabs are more-or-less like the "Sales" tab, they are not repeated here to avoid unnecessary duplications.

Testing

In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error-free and fully meet the requirements outlined in the first step. Three types of testing typically take place: unit testing of individual code modules; integration testing; system testing of the integrated product; and acceptance testing. (Melonfire, 2006)

Unit Testing

Unit testing was done at the class level. It was done by creating test inputs for each class and check the actual output with expected output.

Unit testing is mainly used in components, Database and Client classes, as the Database class contents all the methods to process the SQL queries, and the Client class contents the methods to process how the system handle the user's requests. For example, to check the method public Liability getLiability(String name){}, call the method in a main method and enter the same SQL query to MySQL terminal and check if the two results were same. This system is not include complex calculations and algorithms, there is no extreme case involved, the errors are normally come up with daily uses.

Integration Testing

Integration testing is used for checking how well the classes cooperate with others if any changes in some classes . It is particularly important for Client class, as the Client class is the "bridge" between the *View and Model.* For example, how does it response if there is a change of user's input. Testing is concentrated on getting the view change correctly until they accurately reflected the information processed by the Client and Database classes.

System Testing

Once a full working product was available it was important to test how well the software worked and how well it met the requirements. The system testing was carried out some by some members of the Computer Science department and some friends, and the developer through the GUI. The results of this testing are attached in Appendix 4. There were some errors coming out during the testing, most are the number format and date format issue, the system have no ability to detect this kind of error, but basically most of the requirements were successfully achieved.

Acceptance Testing

The acceptance testing was not carried out due to the distance matter. The potential users which are the interviewers, the small business owner in Linyi, China, consider the difficulty of setting up the database and their computer skills, the acceptance testing is replaced by system testing.

Evaluation

The testing above was very useful to detect the errors and bugs of the software. However, as mentioned above, the testing was carried by some students at School of Computer Science, they are not very familiar with the business activities in realistic. In this evaluation section I use my knowledge as a business student, and put myself as a normal user to evaluate this product.

Vs. Original Specification

After testing and modification, the product meets all of the essential requirements, and also partly achieved the desirable functions. After reconsider the "User management" function will be removed. Because all other functions are the same, every user see the same data and information, so there is no need for distinguishing the root user and non-root user, and the Setting tab for managing users will be changed. The system will pre set the username and password to "admin" and "admin", after login to the system, user can change the username and password at the Setting tab. Although, the system will still need a username and password combination to login, so that will guarantee the security to a certain extent.

The desirable function "News subscription" is only showing the headlines of each stories. It's better to attach the hyperlinks of the headlines to allow the users to see the whole stories. Also, it is showing the news from BBC only, there should be a function to allow users to add their personal subscriptions.

The desirable function "reminder" is only accepting viewing and adding reminders, there is no way to delete them.

Interface Evaluation

Sommerville (2007) recommends that an interface evaluation should be based on five usability attributes, of which the relevant four are given below. These attributes are the reflection of the program as a whole, not just the interface. He also recommends that questionnaires be used as a simple technique for evaluating these attributes. The comments and feedback are collected from the members who did the system testing at School of Computer Science.

Learnability

Most of them were satisfied with the GUI design, and believed the software was easy for use. The biggest issue was the method of display the transactions' records, many of them suggest change the pain text to table style.

Speed of operation

As for the speed of operation, it is a light weight software and the database is in local computer, so the speed should not be a issue, and the fact is there was not any comments about the operation speed. The speed is performing as excepted.

Robustness

Robustness is for evaluate how tolerant the software is from user errors. This was done with the system testing, the results are attached in Appendix 4. From the results, we can see the software can deal with the error inputs correctly.

Recoverability

Recoverability is for evaluate how well the system recovers from user errors. This was done with the system testing, the results are attached in Appendix 4. From the results, we can see the software normally shows warring messages to the user in case of error inputs.

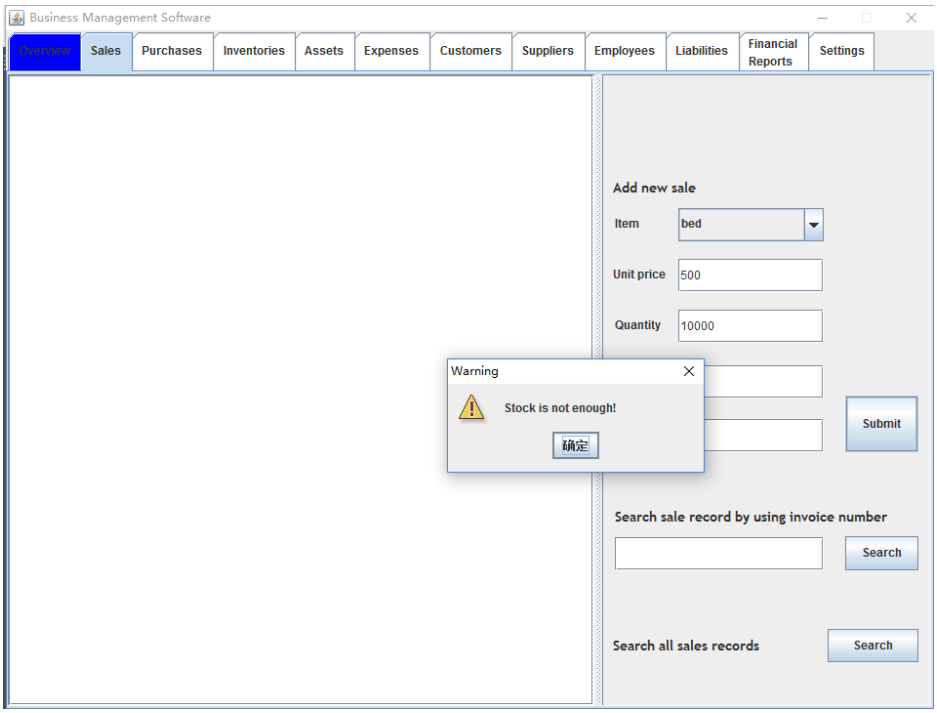


Figure 14: The screen shot of the Warning message for not enough stock.

Discussion

I was a business student before, the best thing for doing this project is to combine my business knowledge and computer science knowledge together and put it in practice. The biggest challenge for me during the project period is time limitation. I was one of the deferred project group, I supposed to start the project in late September or early October, but I started one month later due to the change of supervisor. Luckily, I finished the project on time even there are still many deficits.

Basically, the software achieved all the essential requirements as requested, due to the time matter, many improvements can not be done. If I had more time, I would have done the changes below:

1. Change the transaction records text into a table, to make it more clear.
2. Add a function for users to change the news provider.
3. Allow users to delete reminders.
4. Make the financial statements more specific, for example, in the expenses of income statement, there are detailed list of the expenses.
5. Make the software more intelligent, for example, automatically calculate the assets depreciation.
6. Make the database tables more rational.

Conclusion

Overall the project has been a success. After the problem has been determined, the project followed the waterfall model to solve it step by step, it gave a clear objective at each step, the waterfall model helped a lot during the development process.

Back to the original problem, more than 80% of small business owners in the city which has the second largest wholesale market in China are still using the traditional method to do daily work --- using paper and pen to record all the transactions and other information, and importantly, most of them thought the computer software will have better security, better performance and more efficient, etc. By using this software, users can securely login, clearly see the payables, receivables, monthly sales even financial statement to understand how the business doing. The transactions and other information will be stored into database accordingly, if user want view a past sale record, all he need is type in the invoice number in to the system instead of looking from piles of books.

As mentioned above, due to the time matter there are still deficits of the system. The software will be updated in the future, and it will be available for the interviews of my questionnaires for free.

References:

Overview of Linyi Trade City. (2016) *Linyi Trade City Administrative Commission*.
Available online at: http://en.linyimart.gov.cn/info/1095/2282.htm

Melonfire, C. (2006) Understanding the pros and cons of the Waterfall Model of software development, *TechRepublic*
Available online at: http://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/

Tencent (2014) About Tencent.
Available online at: http://www.tencent.com/en-us/at/abouttencent.shtml

Burbeck, S. (1992) Applications Programming in Smalltalk-80:How to use Model–View–Controller
Available online at: http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html

Beneke, T. and Wieldt, T (2013) JavaOne 2013 Review: Java Takes on the Internet of Things. *Oracle*
Available online at: http://www.oracle.com/technetwork/articles/java/afterglow2013-2030343.html

Sommerville, I. (2007). *Software Engineering.* Addison Wesley: Harlow.

About the code:

The “RSSReader. java” inside the package GUI was taken from Francisco Iacobelli
Available online at: https://www.youtube.com/watch?v=xiK-DH74oJg

All the constructors, getters and setter in the classes inside the component package are generated by Eclipse.

The graphical user interfaces are built with help from Eclipse WindowBuilder.

Appendices:

How to run the application:

System requirement:

Any OS with JDK and MySQL database.

1. Create a database called “project” in your MySQL database, then import the database file to your database with both data and structure.
2. Import the Java project which is the folder called “project” inside the bxt520.zip
3. Open \project\src\database\Database.java, change the field variable url to adapt your database setup.
4. Run \project\src\GUI\ProjectStart.java

Please note, you may need to add “mysql-connector-java-5.1.40-bin.jar” to use JDBC, the file is attached in the .zip file

Appendix 1.

The questionnaire with collected data.

There were 16 interviews in total.

Business Management Software Survey

1. Which is your industry? [Single choice]
   - Manufacturing……18.8%
   - Retail……6.3%
   - Wholesale……..75%
   - Other____.....0

2. Your position at the business [Single choice]
   - Business owner……93.8%
   - Normal employee……..6.3%
   - Accountant……...0
   - Other…….0

3. Are you currently doing your daily work by using computer or pan and paper? [Single choice]

Pan and paper…….84.6%
Computer……15.4%

4. What are the advantages you think about the traditional method?[[Multichoice]
   Low cost…….76.9%
   Easy to use……..92.3%
   Efficient……..0
   Other____.....0

5. What are the disadvantages you think about the traditional method? [Multichoice]
   Inefficient……100%
   Easy to lose……..100%
   Not intelligent……..38.5%
   Not environment friendly……92.3%
   Costly……7.7%
   Other____.....0%
6. About the business management system, what functions are you expecting? [Multichoice]
   View and record transactions……100%
   Manage the information about customers, employees and suppliers……50%
   Other…….0

7. About the business management system, what are the concerns you may have? [Multichoice]
   Efficiency.....100%
   Usability.....81.3%
   Portability.....62.5%
   Security and reliability.....100%
   Cost.....31.3%
   Other.....0

8. Overall, what do you think about using the business management system? [Single choice]
   It has good future, people will accept it. 100%
   Prefer the traditional method……....0

Appendix 2.

Details of the database “project”:

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO	PRI	NULL	
date	date	YES		NULL	
quantity	int(11)	YES		NULL	
unitValue	double	YES		NULL	
totalValue	double	NO		NULL	

The assets table is used to store the data of assets.

Field	Type	Null	Key	Default	Extra
firstName	varchar(255)	YES		NULL	
lastName	varchar(255)	NO		NULL	
address	varchar(255)	YES		NULL	
contactNumber	varchar(255)	NO	PRI	NULL	
emailAddress	varchar(255)	YES		NULL	

The customer table is used to store the data of customers.

Field	Type	Null	Key	Default	Extra
EmployeeID	int(11)	NO	PRI	NULL	auto_increment
LastName	varchar(255)	NO		NULL	
FirstName	varchar(255)	YES		NULL	
Address	varchar(255)	YES		NULL	
ContactNumber	varchar(255)	YES		NULL	
Gender	varchar(255)	YES		NULL	
Age	int(11)	YES		NULL	
DateOfJoin	date	YES		NULL	

The employees table is used to store the data of employees.

Field	Type	Null	Key	Default	Extra
no	int(11)	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	
amount	double	YES		NULL	
date	date	YES		NULL	

The expenses table is used to store the data of expenses.

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO	PRI	NULL	
date	date	YES		NULL	
quantity	int(11)	YES		NULL	
unitCost	double	YES		NULL	
totalCost	double	YES		NULL	

The inventories table is used to store the data of inventories.

Field	Type	Null	Key	Default	Extra
item	varchar(255)	NO	PRI	NULL	
amount	double	NO		NULL	
date	date	NO		NULL	

The liabilities table is used to store the data of liabilities.

Field	Type	Null	Key	Default	Extra
note	varchar(255)	NO		NULL	
date	date	YES		NULL	
id	int(11)	NO	PRI	NULL	auto_increment

The notes table is used to store the data of notes.

Field	Type	Null	Key	Default	Extra
name	varchar(255)	NO	PRI	NULL	
date	date	YES		NULL	
amount	double	YES		NULL	

The payables table is used to store the data of payables.

Field	Type	Null	Key	Default	Extra
invoiceNumber	int(11)	NO	PRI	NULL	
name	varchar(255)	NO		NULL	
unitPrice	double	YES		NULL	
quantity	int(11)	YES		NULL	
totalPrice	double	YES		NULL	
supplier	varchar(255)	YES		NULL	
dateOfPurchase	date	YES		NULL	

The purchases table is used to store the data of purchases.

Field	Type	Null	Key	Default	Extra
invoiceNumber	int(11)	NO	PRI	NULL	
customer	varchar(255)	YES		NULL	
date	date	YES		NULL	
amount	double	NO		NULL	

The receivables table is used to store the data of receivables.

Field	Type	Null	Key	Default	Extra
product	varchar(255)	NO	MUL	NULL	
invoicenumber	int(11)	NO	PRI	NULL	auto_increment
unitprice	double	YES		NULL	
quantity	int(11)	YES		NULL	
totalPrice	double	YES		NULL	
VAT	double	YES		NULL	
VATRate	double	YES		NULL	
date	date	YES		NULL	
customer	varchar(255)	YES		NULL	

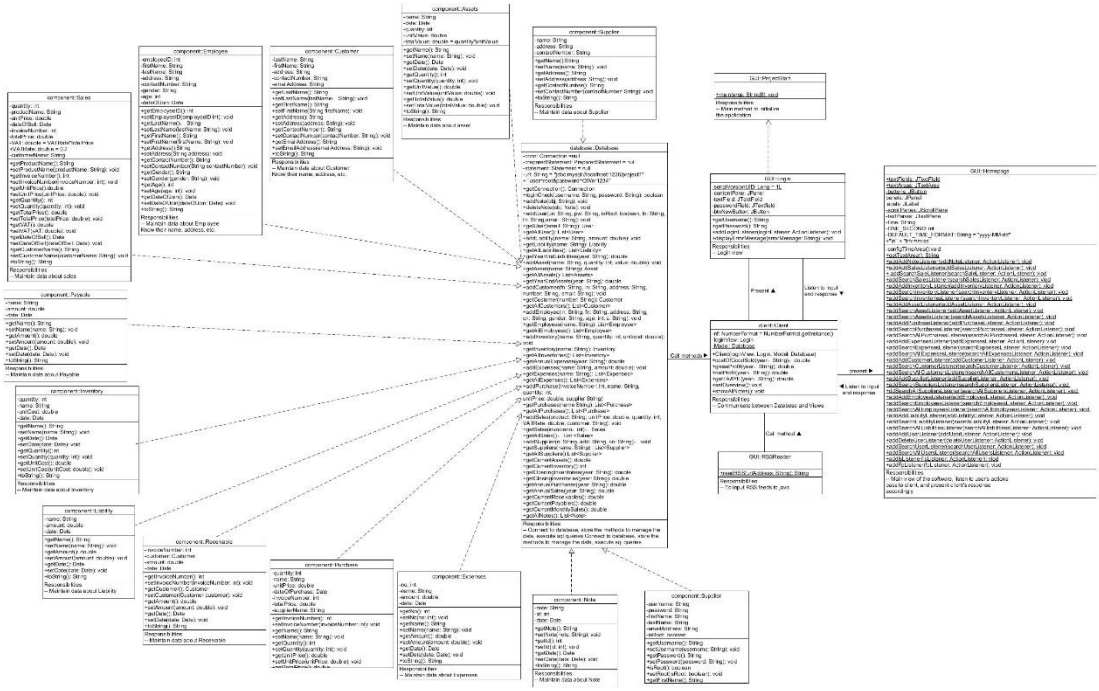
The sales table is used to store the data of sales.

Field	Type	Null	Key	Default	Extra
name	varchar(255)	YES		NULL	
address	varchar(255)	YES		NULL	
contactNumber	varchar(255)	NO	PRI	NULL	

Field	Type	Null	Key	Default	Extra
username	varchar(255)	NO	PRI	NULL	
password	varchar(255)	NO		NULL	
firstName	varchar(255)	YES		NULL	
lastName	varchar(255)	YES		NULL	
emailAddress	varchar(255)	YES		NULL	
isRoot	tinyint(1)	NO		NULL	

Appendix 3.

Class diagram.



The original sized class diagram photo is attached in the .zip file.

Appendix 4.

The system test results.

Feature	Test	Expected	Outcome
Login check	Leave the username or password or both blank	Error dialogue	As expected
	Username or password or both wrong	Error dialogue	As expected
	Username and password are correct	Login and to main window	As expected
Reminder	Add a note	Display the note added in the textArea	As expected
	Delete a note	No way to delete	As expected
Overview display update	Add a sale	Current sales in this month increase by the new sale amount	As expected
	Add a receivable	Total amount of receivables increase by the new receivable amount	As expected
	Add a payable	Total amount of payables increase by the new payables amount	As expected
Sales tab test	Add a sale but leave unit price or quantity or vat rate blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
	Type a larger amount than current stock	Error dialogue	As expected

Purchases tab test	Add a purchase but leave some boxes blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
Inventories tab test	Add a inventory but leave item or quantity blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
	Add a inventory which is not in the current list	Dropbox at the item of sales tab add the new inventory	As expected
Assets tab test	Add a asset but leave some boxes blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
Expenses tab test	Add a expense but leave some boxes blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	Not error detected
Customer tab test	Add a customer but leave some boxes blank	At least fill last name and contact no. otherwise error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
Suppliers tab test	Add a supplier but leave some boxes blank	At least fill name and contact no. otherwise error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected

Employees tab test	Add a employee but leave some boxes blank	At least fill last name and contact no. otherwise error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
	Type wrong date format inside the date of join	Error dialogue	No error detected
Liabilities tab test	Add a liability but leave some boxes blank	Error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	Not error detected
Financial Report tab test	Type year in the boxes	Show the statements	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected
Settings tab test (only for root user, non-root user won't see settings tab)	Add a liability but leave some boxes blank	At least fill username, password, root user and email otherwise error dialogue	As expected
	Type non-numeric inside a numeric required boxes	Error dialogue	No error detected