Design and Implementation of WPF-based Mini OA System

Cheng WANG *, Zhixiao LIN, Fangsheng GUOTANG, Lixiao ZHENG, Zhigang ZHU
College of Computer Science and Technology
HuaQiao University, Xiamen 361021, China
wangcheng@hq.edu.cn

ABSTRACT: *In order to build a practical mini OA system for small and medium enterprise, key business, overall requirement, functional requirements and non-functional requirements of Yardi Kooboo (Xiamen) Technology Co.Ltd. are analyzed after understanding the company's specific workflow and management processes. Following software engineering, the entire progress of building a WPF-based OA system includes functional module design, database design, achievement of three-tier architecture, achievement of mail cache and so on. Deployment of this new OA system improves the level of information management and obtains good application effect. The development process provides good reference value for the development and research of similar systems.*

Keywords: Office Automation System, WPF, C/S, Software Engineering, System Modularity

Received: 19 June 2015, Revised 20 July 2016, Accepted 28 July 2016

1. Introduction

In the information age, computer is helpful for office staff in processing information, for it has greatly improved the efficiency of individual information processes [1]. When it comes to modern office automation age, the goal of office automation is not only limited to lighten the workload of individual employees, but to let the computer handle cumbersome things, so that employees can focus on the most important thing, and at the same time, also improve efficiency of team work, thereby improve the overall efficiency of enterprises and strengthen the overall competitiveness. In this work mode, OA system [2] can quickly spread one's mind to other people, and different departments can collaborate on the basis of shared information. So every department and even individual will clearly know their responsibility and receive feedback quickly, then decision-maker can quickly integrate information from various aspects, and make their own decisions fairly, objectively and accurately [3]. With the rapid development of information and network technology, each company inevitably requires suitable office automation (OA) system to manage information efficiently.

The development process of OA system can roughly be divided into three generations according to the historical development of OA [4]: (1). Beginning of OA system. In the 80s of last century, with the development of personal computers and other technologies, electronic document writing and statistics progresses change from paper format to electronic digital ones. (2). Development of collaborative OA system. Its features include network technology, collaborative work and workflow automation. (3). Collaboration and knowledge-based OA systems. The third generation OA system will upgrade the second generation OA system to knowledge-managed core level [5, 6].

However, OA system requirements of each company are different. Big companies need large, comprehensive and high-level OA system. Because of complex requirements of software structure, and high time-cost development and maintenance process, they tend to choose large software companies to develop [7]. Referring to its high price and function, such large and comprehensive OA system is not suitable for small companies. They need a small and specialized OA system [8]. To meet the demand of small and medium companies, the design and implementation process of SMEs (small and medium enterprise) OA system with low cost is studyed in this research.

Kooboo (Xiamen) Technology Co., Ltd. is a software company, committing to research and development computer software with high-tech, mainly engaged in e-commerce website development, search engine optimization, website promotion and outsourcing services, etc., which owns strong research and development capabilities and extensive customer group. The company owns 60 employees in China, and also 20 employees aboard. Based on actual OA demand of this company, the whole process of building a WPF-based small and medium enterprise OA system for collaborative work system following software engineering has higher more practical value. This system will eventually meet Kooboo's demand. Requiring the document placed to be under the system permission restrictions and be preserved as perfect as possible, the system could make the work orderly and timely, and really improve the work efficiency and quality of the small and medium companies.

2. OA System Requirements Analysis

2.1 Key Business Analysis

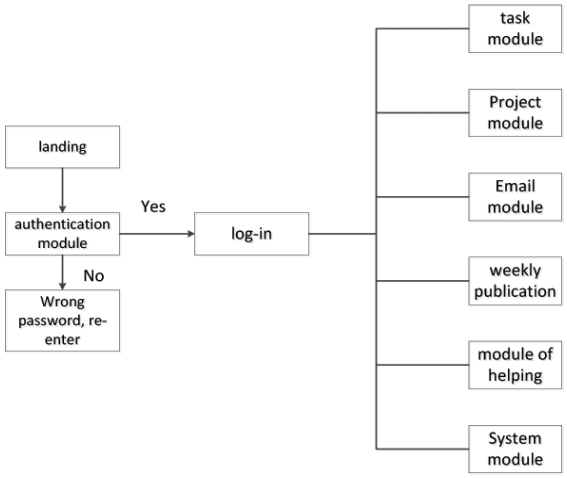


Figure 1. Flow chart of system operation

The main function of this system is to select the appropriate module for processing information after log-in through the login module. Flow chart of system operation is shown in Figure 1.

2.2 Overall Requirements Analysis

Before design and development of a software system, requirements must be understood clearly firstly. Services to be provided and the description of the constraints are what the system requirements concern. The process of discovering service, analyzing and establishing documentation, and inspecting are requirements engineering. Software requirements cycle analysis is the most critical step in software life.

In order to meet the general demand, the following functions must be included in the OA system:

- **Nice UI Interface**

Interface (beautiful or not) relates to the user experience. Nice user interface can not only bring a good mood, but also enhance the user's interaction with the system.

- **Suitable Permission Constraint for Kooboo**

The OA system must own permission constraint management function for Kooboo. Since this system is specially developed for Kooboo, the permission constraint management system must comply with Kooboo's actual situation, which will enable employees to use it handily.

- **Reflect Superiority of the OA System in the Task Management and Project Management**

Task management and project management are the top priority in this system, which must be noted in the design and development of the system, reflecting differences between the OA system and other systems.

- **Good Data Processing Efficiency**

The OA system is for internal use, and the database uses SQL SERVER 2008. During the process of system design and development, it must be cared to optimize so that the system has better processing efficiency, which can enhance the user experience of system.

2.3 Functional Requirements Analysis

2.3.1 Task Management Functions

Task management function is one of the core modules of the system. When the user is authenticated, he or she can see all related tasks. There are several small independent, but interdependent and mutual cooperativetask sub-modules in the task module . The functions of task sub-modules are listed as below.

- Start, pause and submit task: the user can carry the above operations out according to the tasks assigned to them.
- Add, delete and modify task: it is used for details exchange for the task. User with the specified permissions can perform these operations.
- Task activity: In the start, pause, submit, change and auditing tasks, the system will automatically record the activities the users perform about the task.
- Tasks time: The system will automatically record the time user spent on the task.
- Task-related e-mail: In this system, multiple messages can be associated with a task, which can be more convenient to view.

Use case diagram of task module is shown in Figure 2.

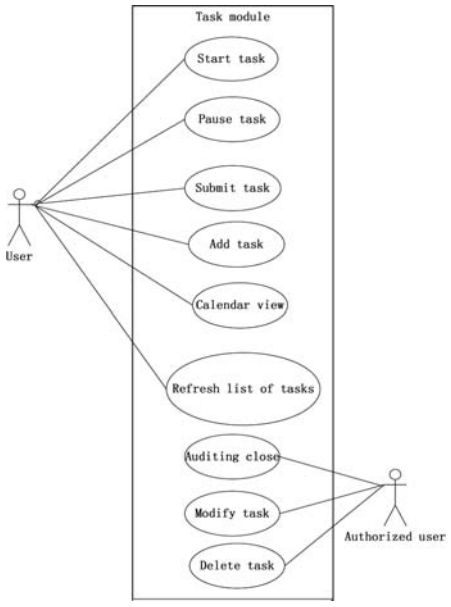


Figure 2. Use case diagram of task module

2.3.2Authentication

The main function is the authentication when user enters the C/S or B/S platform and permissions access when the work carried out on the platform. Authentication can effectively prevent the leakage of important information of company.

2.3.3 Project Management Capabilities

Project management function is also one of the core modules of the system. Project module is closely linked to task. There are many tasks in a project, and assigning a new task also needs to select a project and project members. The main functions of the project are constructing project, modifying the project and ending/ starting the project. Users' permissions when log-in determines whether the user has permission to carry out the above operations. Use case diagram of project module is shown in Figure 3.

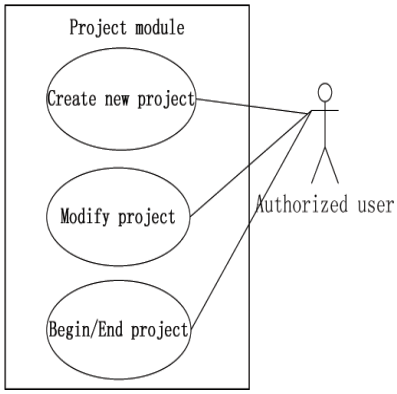


Figure 3. Use case diagram of project module

2.3.4 Weekly Feature

The function of this module is to show the logined user's work week duration and the daily tasks and her/ his completed tasks of corresponding number of weeks, etc.. And it allows the user to confirm the correct length of work time the week etc. Use case diagram of weekly feature module is shown in Figure 4.

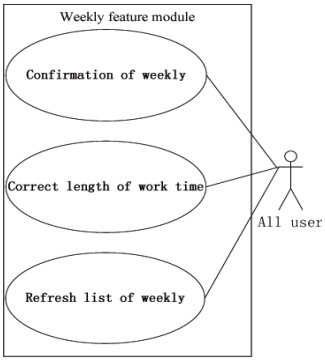


Figure 4. Use case diagram of weekly feature module

2.3.5 Systems Help Function

The role of module is to provide a certain type of task as standard reference. Each user can publish their own help documentation which can help new employees greatly. Use case diagram of systems help function module is shown in Figure 5.

2.4 Non- Functional Requirements Analysis

2.4.1 Performance Requirements

1). Response rate: all kinds of operation response time <1s.

2). To meet the visitors< 200.

3). Occupy system resources <200M.

2.4.2 Security Requirements

1). Irreversible MD5 encryption is adopted by users' login password.
2). System permission constraint..
3). Users can only access the information they can access to.

3. Functional Module Design And Database Design

3.1 Functional Module Design

3.1.1 Landing Module Processes

Landing is one of the needed functions of an enterprise OA system. Landing page asks for the user entering a user name and password, then the client will enter user and password for authentication. If the user name and password are correct, the client enters the login state, after login, the system queries the corresponding database table under the user permissions. If the username or password is incorrect, the system will display an error message. Flow chart of landing module is shown in Figure 6.

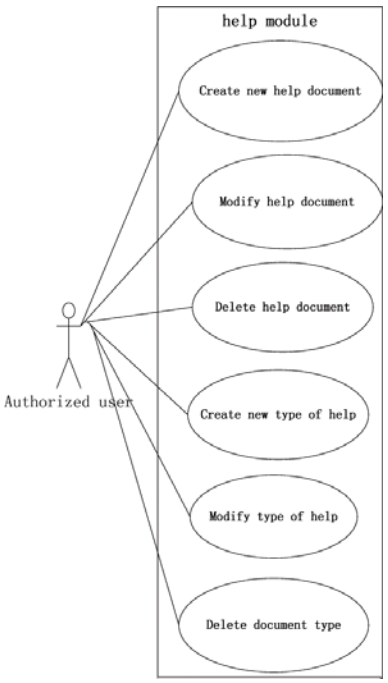


Figure 5. Use case diagram of systems help function module

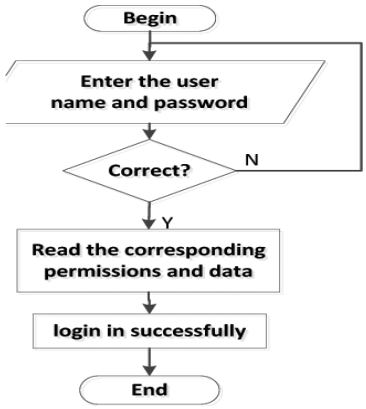


Figure 6. Flow chart of log-in module

3.1.2 Task Module

Task module is one of the most important modules of OA system. When a user receives an assigned task, he begins the task. At the beginning of the task, regarding complication of the task, he can pause the task or submit the task. If the task is uncompleted, it can be suspended until next time. If it has been completed, the task can be submitted. After that, the task can be assigned to others to deal with. For example, after a module developed by developers, it can be assigned to testers for testing then. It can also be submitted directly to wait for other users with the audit authority to audit. If the user has audit authority, then he can carry on the task the audit. If the test is passed, then this task is end; if not passed, he can write down the reason for people to continue the development process. Flow chart of task module is shown in Figure 7.

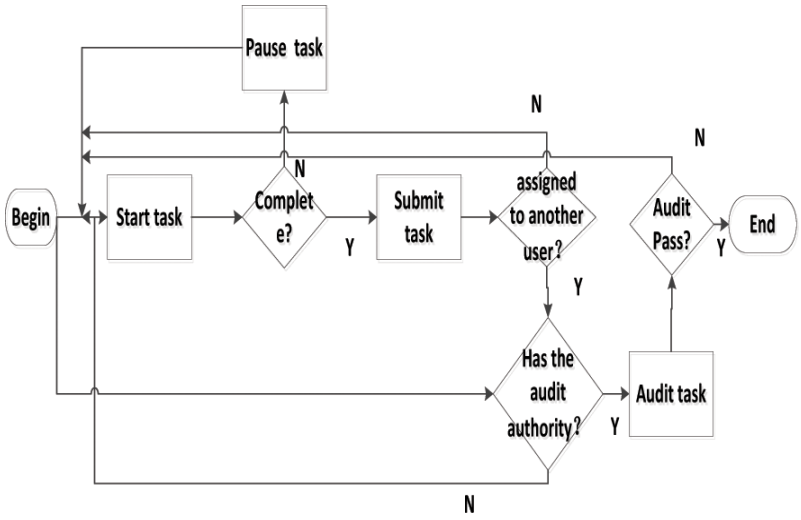


Figure 7. Flow chart of task module

3.2 Database Design

3.2.1 Logical Structure of the Database Design

Usually in the process of designing a database, the first thing is analyzing requirements, using conceptual model (ER model or object-oriented model) design pattern based on demand , then turning conceptual model into the data model .

There are several design principles during the design process of ER model: loyalty, avoiding redundancy, simplicity, contact choices and choosing the right kind of element.

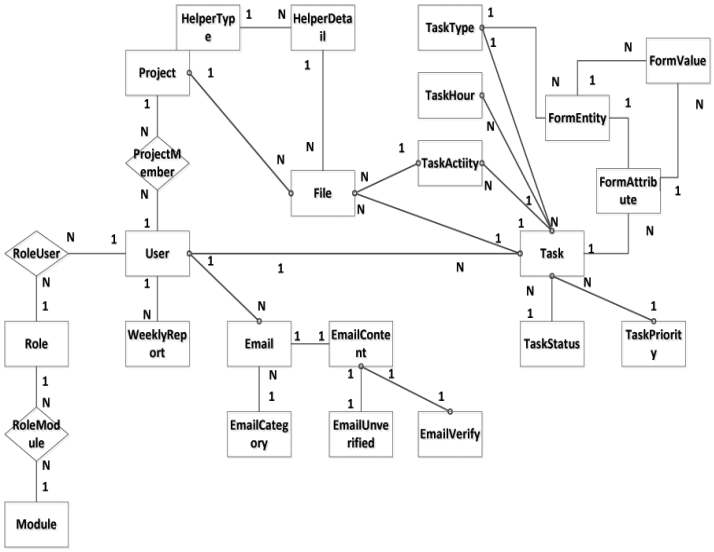


Figure 8. Database E-R diagram

The above database design process must be followed. After defining database demand clearly, the ER diagrams based on database need designing. In the process of designing ER conceptual model, several ER designed principles should be followed to make relation normalized, so that it is easy for database operations. ER diagram according to the database demand is shown in Figure 8.

3.2.2 Database Table Relationship Diagram

According to ER diagram in Figure 8, the actual database design model which means relations between each table should be realized, as shown in Figure 9.

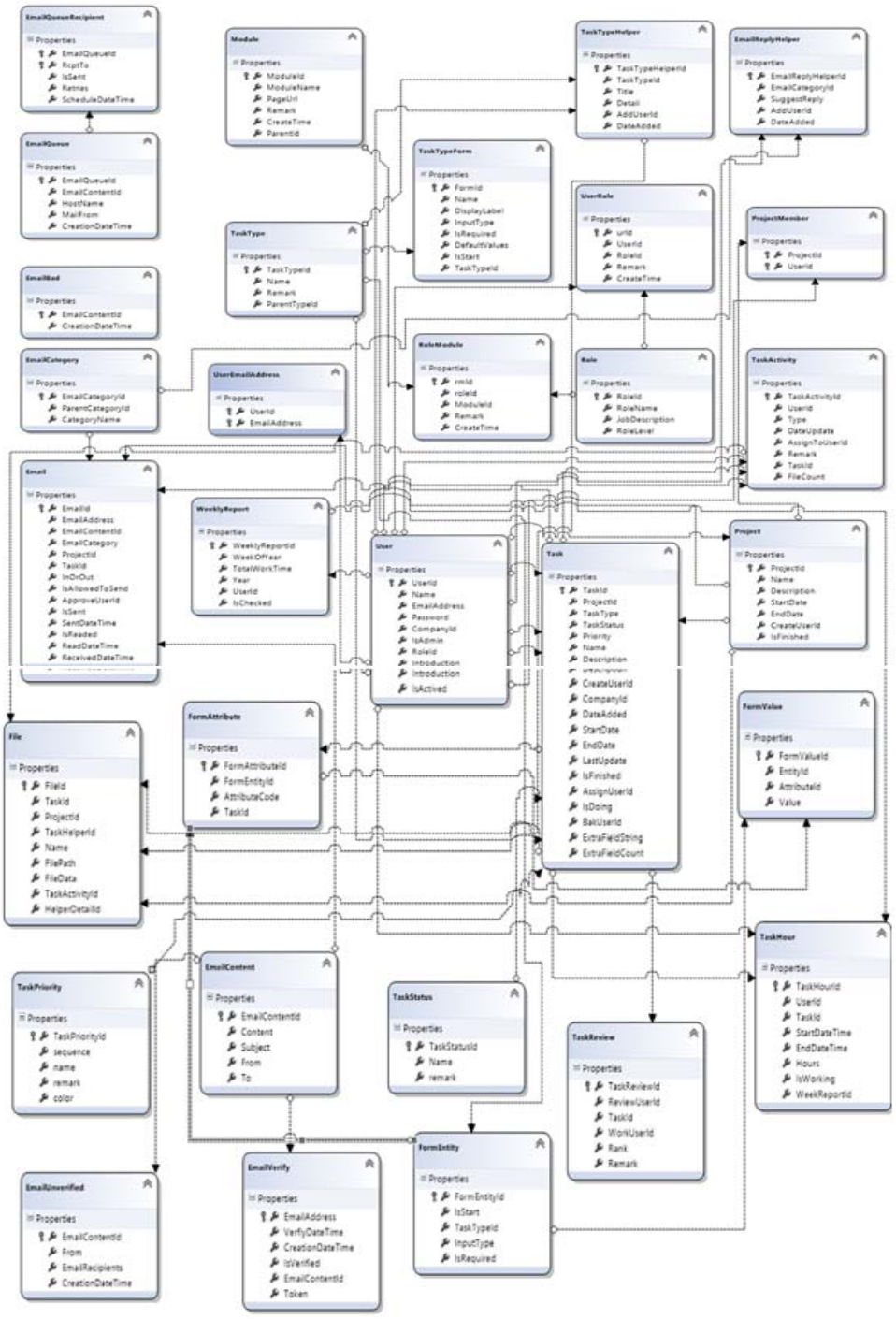


Figure 9. Database table relationship diagram

4. OA System Implementation

4.1 Achievement of Three-tier Architecture

Various modules of the system are implemented with the classic three-tier architecture. Achievement of database access three-tier architecture by WPF is taken for example.

4.1.1 Data Access Layer

Data Access Layer takes linq data access object instance TaskDataContext. Current to obtain data. The instance TaskDataContext.Current is obtained in the place as presented in Figure 10, and sustenance in the thread, so that the entire program can use this object to perform data access, and to avoid database connection error result from excessive lead.

```
public static TaskDataContext Current
{
    get
    {
        var result = Kooboo.CallContext.Current.GetObject<TaskDataContext>("TaskCtx");
        if (result == null)
        {
            result = CreateContext();
            Kooboo.CallContext.Current.RegisterObject<TaskDataContext>("TaskCtx", result);
        }
        return result;
    }
}
```

Figure 10. Code of TaskDatacontext.Current object instantiation

After getting linq data access object instance, taking the example of linq can automatically generated table objects for data query and returns IQueryable list. Linq does not really remove the data when generating SQL to query the database statement, but only to query in the business logic layer assignment.

```
[global::System.Data.Linq.Mapping.TableAttribute(Name="dbo.Task")]
public partial class Task : INotifyPropertyChanging, INotifyPropertyChanged
{

    private static PropertyChangingEventArgs emptyChangingEventArgs = new PropertyChangingEventArgs(String.Empty);

    private int _TaskId;

    private int _ProjectId;

    private System.Nullable<int> _TaskType;

    private System.Nullable<int> _TaskStatus;

    private int _Priority;

    private string _Name;

    private string _Description;

    private int _CreateUserId;

    private int _CompanyId;

    private System.DateTime _DateAdded;

    private System.DateTime _StartDate;

    private System.DateTime _EndDate;
```

Figure 11. Code of linq objects generate Task Sheet

4.1.2 Business Logic Layer

Business logic layer is mainly for business logic operations on the data layer and the business logic processing. Take a simple

business logic processing for example. IssueMainPage carries on business logic processing at the initialization time. As shown in Figure 13, when initializing the page, the first thing is to read resource and carry out its instantiation such as all controls inon beside the XAML pageto. The second thing is to call the InitCombobox method to initialize all the Combobox. The next thing is to call the loadingTaskService method after the page is complete read. The static method created in data access layer is called for data access, and then the binding method is called to bind data to bind the data to the interface.

```
public static class TaskBiz
{

    public static IQueryable<Task> GetAllTask()
    {
        return TaskDataContext.Current.Task.OrderBy(o => o.IsFinished)
                .ThenByDescending(o => o.Priority).ThenBy(o => o.EndDate);
    }

    public static IQueryable<Task> FilterByUserId(this IQueryable<Task> list, int userId)
    {
        var query = list.Where(o => o.Project.ProjectMembers.Select(m => (int)m.UserId).Contains(userId) &&
            o.IsFinished == false)
            .OrderByDescending(o => o.Priority).ThenBy(o => o.EndDate);
        return query;
    }


    public static IQueryable<Task> FilterByUserIdWithFinished(this IQueryable<Task> list, int userId)
    {
        var query = list.Where(o => o.Project.ProjectMembers.Select(m => (int)m.UserId).Contains(userId))
            .OrderBy(o => o.IsFinished)
            .ThenByDescending(o => o.Priority).ThenBy(o => o.EndDate);
        return query;
    }
```

Figure 12. Code of data access layer part

```
public IssueMainPage(int projectId)
{
    InitializeComponent();
    ProjectId = projectId;

    InitCombobox();

    taskBtnProperty = new BindingTaskBtnProperty();

    this.Loaded += (s, e) =>
    {
        loadingTaskService();
        TaskDetailTabControl.SelectionChanged += TaskDetailTabControl_SelectionChanged;
    };
}

private void loadingTaskService()
{
    TaskDataContext.Clear();
    //普通导航到Task页面
    if (ProjectId == -1)
    {
        //Issues.Clear();
        var allAboutTask = TaskBiz.GetAllTask();
        Binding(allAboutTask, ConstId.TaskPageNumber, 1, 0);
    }
    else
    {//从查看对应项目的任务
        var tasks = TaskDataContext.Current.Task.FilterByProjectId(ProjectId);
        Binding(tasks, ConstId.TaskPageNumber, 1, 0);
    }
}
```

Figure 13. Code of IssueMainPage initialized business logic processing

With the Binding method, the newly created thread is used to load the data, then to delegate the data obtained to the UI thread Dispatcher page for UI updates. The specific procedure is shown in Figure 14.

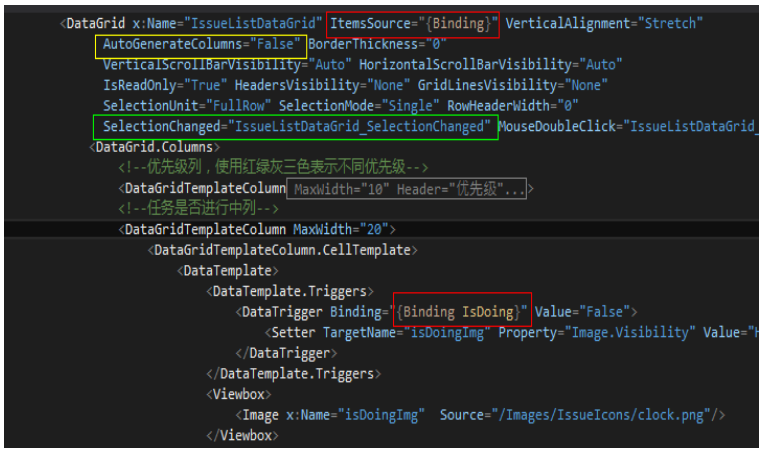
```
public void Binding(IQueryable<Task> tasks, int number, int currentSize, int selectedIndex)
{
    this.ListGrid.IsEnabled = false;
    this.IssueListDataGrid.Foreground = new SolidColorBrush(Colors.Gray);
    this.progressBar.Visibility = Visibility.Visible;
    lock (tasks)
    {
        new Thread((ThreadStart)delegate
            {
            Dispatcher.Invoke((ThreadStart)delegate
            {
                #region main
                //总个数
                int totalNumber = tasks.Count();
                //总页数
                int totalPage = 0;
                if (totalNumber != 0)
                {
                    if (totalNumber % number == 0)
                    {
                        totalPage = totalNumber / number;
                    }
                    else
                    {
                        totalPage = totalNumber / number + 1;
                    }

                    var newIssueList = tasks.Take(number * currentSize).Skip(number * (currentSize - 1));
                }
```

Figure 14. Code of binding method

4.1.3 Presentation Layer

In WPF, presentation layer is written with a markup language XAML. A piece of XAML code is listed as shown in Figure 15



```
<DataGrid x:Name="IssueListDataGrid" ItemsSource="{Binding}" VerticalAlignment="Stretch"
    AutoGenerateColumns="False" BorderThickness="0"
    VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Auto"
    IsReadOnly="True" HeadersVisibility="None" GridLinesVisibility="None"
    SelectionUnit="FullRow" SelectionMode="Single" RowHeaderWidth="0"
    SelectionChanged="IssueListDataGrid_SelectionChanged" MouseDoubleClick="IssueListDataGrid_
    <DataGrid.Columns>
        <!--优先级列，使用红绿灰三色表示不同优先级-->
        <DataGridTemplateColumn MaxWidth="10" Header="优先级"...>
        <!--任务是否进行中列-->
        <DataGridTemplateColumn MaxWidth="20">
            <DataGridTemplateColumn.CellTemplate>
                <DataTemplate>
                    <DataTemplate.Triggers>
                        <DataTrigger Binding="{Binding IsDoing}" Value="False">
                            <Setter TargetName="isDoingImg" Property="Image.Visibility" Value="H
                        </DataTrigger>
                    </DataTemplate.Triggers>
                    <Viewbox>
                        <Image x:Name="isDoingImg"  Source="/Images/IssueIcons/clock.png"/>
                    </Viewbox>
```

Figure 15. Code of task list XAML

Here DataGrid is used as the task list. In the DataGrid, codes similar to red selected are used to bind data to the interface, while the yellow selected code acts as the specified property, and the green selected code bind the event to the background code.

4.2 Achievement of Mail Cache

Three-tier implementation of e-mail module will not be repeat here. Because mail module is different from other modules, the message itself may be large. E-mail module without local cache will greatly affect the user experience, so a local cache e-mail system is need to be designed.

Local structure of the e-mail cache is shown in Figure 16.

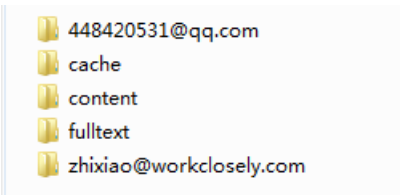


Figure 16. E-mail local cache file structure

In figure 16, the folder named as Mail addresses is personal mail folder cache. There are two folders called in and out representing inbox and outbox cache of email. Inside the folder is the header department of message cache, and the file name is the ID of the message in the database. When loading the mail list, the system only loads the message header, which greatly reduces the loading time and memory, as shown in Figure 17.

```
Received: by self with self;Fri, 28 Dec 2012 07:37:14 +0000
Subject: test
From: 09计科林志霄 <448420531@qq.com>
To: <448420531@qq.com>
Content-Type: multipart/alternative;boundary=502c5c5088184a0b9af67a4fcc570b1c
MIME-Version: 1.0
Date: Fri, 28 Dec 2012 07:37:14 +0000
Message-ID: <eda24dfa-903c-42eb-866f-86c9a3a7cc6d@Suning-PC>
X-Readed: True
X-TaskId: 0
```

Figure 17. Email headers cache screenshot

In the cache folder named as ID contains all local mail cache, each folder contains the attachments of message and message content. When the system is in need of getting a specific message content, it loads the contents of the corresponding folder based on the ID. Mail cache screenshot is shown in Figure 18.

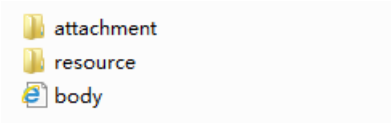


Figure 18. Mail cache screenshot

Source code of the message which is message text without parsed is put into Content folder, as shown in Figure 19 and Figure 20.

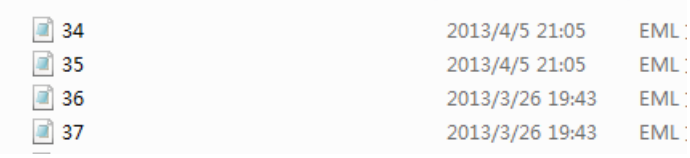


Figure 19. Content folder structure

```
Received: by self with self;Fri, 28 Dec 2012 07:37:14 +0000
Subject: test
From: 09计科林志霄 <448420531@qq.com>
To: <448420531@qq.com>
Content-Type: multipart/alternative;boundary=502c5c5088184a0b9af67a4fcc570b1c
MIME-Version: 1.0
Date: Fri, 28 Dec 2012 07:37:14 +0000
Message-ID: <eda24dfa-903c-42eb-866f-86c9a3a7cc6d@Suning-PC>
X-Readed: True
X-TaskId: 0

--502c5c5088184a0b9af67a4fcc570b1c
Content-Type: text/plain;charset=utf-8


1234
--502c5c5088184a0b9af67a4fcc570b1c
Content-Type: text/html;charset=utf-8


<DIV><P><SPAN>1234</SPAN></P></DIV>
--502c5c5088184a0b9af67a4fcc570b1c--
```

Figure 20. Original structure of the Mail

Fulltext folder is designed to be more efficient and rapid when searching, to improve the user experience. The modules of original text messages, which can be used for searching, are put into Fulltext folder together, making the search more quickly. The structure of fulltext file is shown in Figure 21.

```
adsadfasdf
Zhixiao <zhixiao@workclosely.com>
448420531@qq.com; zhixiao@workclosely.com
23:03:10
asdfsadfsadf
```

Figure 21. Structure of fulltext file

5. Conclusion and Future Research Direction

This system is built based on the specific needs of Yardi Kooboo (Xiamen) Technology Co.Ltd after understanding its specific workflow and management processes. From design of interface to constructing the system, the purpose of building practical, convenient and beautiful system for the convenience of office staff and improvement of work efficiency,is considered.

The development process provides good reference value for the development and research of similar systems. Features of the system are summarized as follows:

1. This system is designed and developed based on the actual needs of Yardi Kooboo (Xiamen) Technology Co.Ltd, and achieved good results in the actual operation.

2. This system runs on multiple systems (Windows 7/Windows 8/Window Server 2008). Database is currently the most stable version of SQL SERVER: SQL Server 2008.

3. Making the daily work flow methodical, and record each person's work, making management decisions more just and equitable. At the same time, dates, type, etc. using drop-down box to display the data of time, dates, type, etc., so that the user can just click the mouse to input, minimizing unnecessary input errors. This can not only reduces the error occurred, but brings users convenient.

4. Timely report to the user when an exception occurs.

5. Use the Ribbon interface, user-friendly interface.

However, there are still some shortcomings existing in this system, including:

1. Since the database hosted elsewhere, making performance influenced

2. The system uses Linq to Sql as database connections, when data in the database is overlarge, the performance of the system will be affected to some extent

3. The design of the database still owns some redundant field, which need to be further optimized

4. The code is too long and in some places relatively strong coupling, which is not conducive to future maintenance and reconstruction.

The system needs to further revise and improve and the authors will continue to strive to the knowledge for further maintenance of the system.

Acknowledgements

This work was financially supported by National Natural Science Foundation of China (Grant No. 51305142) and introduction of talents Huaqiao University Scientific Research Projects (Project No. 12BS217).

References

[1] National Standardization Management Committee. GB / T 5271.27-2001. *Chinese mandatory national standard assembly electronics and information technology* [G255.54 / S]. Beijing : China Standard Press, 2003 .

[2] Kong Lei. (2011). NET-based network OA system design and implementation [TP311.52 / D]. Xi'an Technological University engineering master's degree thesis.

[3] Zhang Yun policy. (2011). H Provincial Hospital Association collaborative OA office system design and implementation [TP311.52 / D]. *Engineering master's degree thesis University of Electronic Science and Technology*.

[4] Sun Min. (2010). office automation [TP317.1 / M]. Beijing Institute of Technology Press.

[5] Nai Chi. (2009). OA system planning and design [TP317.1 / J]. China Science and Technology Information, 15: 128.

[6] Wangzuo Peng., Zhang Yi., Lai Ming. (2005). Cooperative Office Automation System Research and Design [A]. Asia International Process Automation Technology and Equipment Exhibition Proceedings [TP317.1 / C], 2005.

[7] Li Ning. (2011). Safety and efficiency of the OA system design and implementation [TP317.1 / D]. Shandong University master's degree thesis.

[8] Liu G., Aviation Xinghua. (2011). OA System Design and Implementation [TP317.1; TP311.52 / D]. *Engineering master's degree thesis University of Electronic Science and Technology*.