System and Program Design

INTRODUCTION

Inventory Management is one of the basic problems for a

company. It may cause a lot of paperwork, if there is no au-

tomated system available. Implementing such a system is

possible but there are a lot of preliminary works such as de-

termination of the requirements, system structure decision –

software requirements, barcode system selection and deter-

mination of the software tools

INTRODUCTION

Inventory Management is one of the basic problems for a

company. It may cause a lot of paperwork, if there is no au-

tomated system available. Implementing such a system is

possible but there are a lot of preliminary works such as de-

termination of the requirements, system structure decision –

software requirements, barcode system selection and deter-

mination of the software tools

**System Features**

Proposed Database is intended to store, retrieve, update, and manipulate information related to the system which include

1. Order Processing & taking
2. Customer Bill Detail
3. Product Details
4. Calculation of Revenue
5. Searching of product
6. Remainder About Products expiry, Shortage
7. Generate Reports

**Functional Requirements**

* The software must allow input of products data from administrator& secured access at, and from the data streaming real-time monitoring equipment
* The project must request username and password for access to data, only after authentication will allow access to the system.
* The project must require high levels of error correction and input validation.
* The project must allow browsing by the admin to access and update information.
* The project must identify the products by a unique numeric identifier derived from a function on product id.
* The software to be developed must operate without interruption twenty-four hours a day.
* The software to be developed shall display the correct time of day in compliance with standards.
* The software must retrieve, update, and store data from multiple input.
* The software must allow full and complete record search queries by users.

**Non Functional Requirements**

* The software interface must follow design conventions which allow for familiar location of menus, etc.
* Input errors will be returned in red with appropriate message box.
* More than three attempts at login and failure will produce a red flag to system administrator.
* Response time should be minimum.
* No of daily system downs should not more than 10.
* System should Automatically Update after Every Transaction.

**Performance Requirement**

* The software should be able to support at least three simultaneous users.
* 95% of the transactions shall be processed in less than one second.
* Data should be secured and backed up every quarter hour.
* Power supply should have a backup and a disaster recovery plan.
* System should be operable 24 hours a day and accessible in real-time.
* Encryption will Enable Security

**User Requirement**

* The user manual, installation guide and other related material should be sufficient to educate the user how to use and maintain the system.

**Safety Requirements**

* The Database may get crashed or damaged due to some viruses or operating system requirements. Therefore it is mandatory to have backup for your data. Ups/inverter facility should be there in case of power failure.

**Security Requirements**

* System will use secure Database
* Staff can just see and search the products
* Proper user Authentication Will be provided.
* There should be separate account for Admin & user. So that no one else can access the database except Admin.

**Software Architecture**

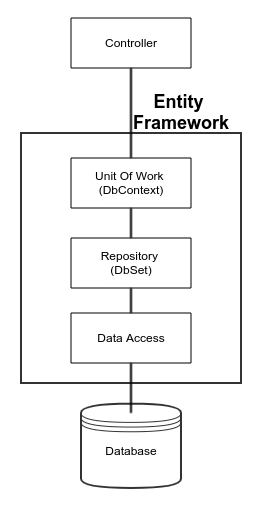
Although we considered much possible architecture, the Repository-Based Architecture seemed fairly appropriate, so we present it here:

**Repository-Based Design**

Repository based architecture is very similar to the client-server model. The system revolves around a central data structure (usually a database), and a collection of independent components which operate on the central data structure. In the case, the central data structure would be the inventory database. Each client would access the server over the network and query the data they require. We decided that a repository-based system would be most appropriate for the software we are trying to design. The central data structure would be the inventory. The database could be designed and installed directly and is instantly accessible from any computer on the network. We felt that there was no need to complicate the situation by creating multiple tiers and having any computer capable of being the service provider. This system also provides a level of security. It also greatly simplifies the construction of the system for the developers and the use of the system for employees.

**Advantages**

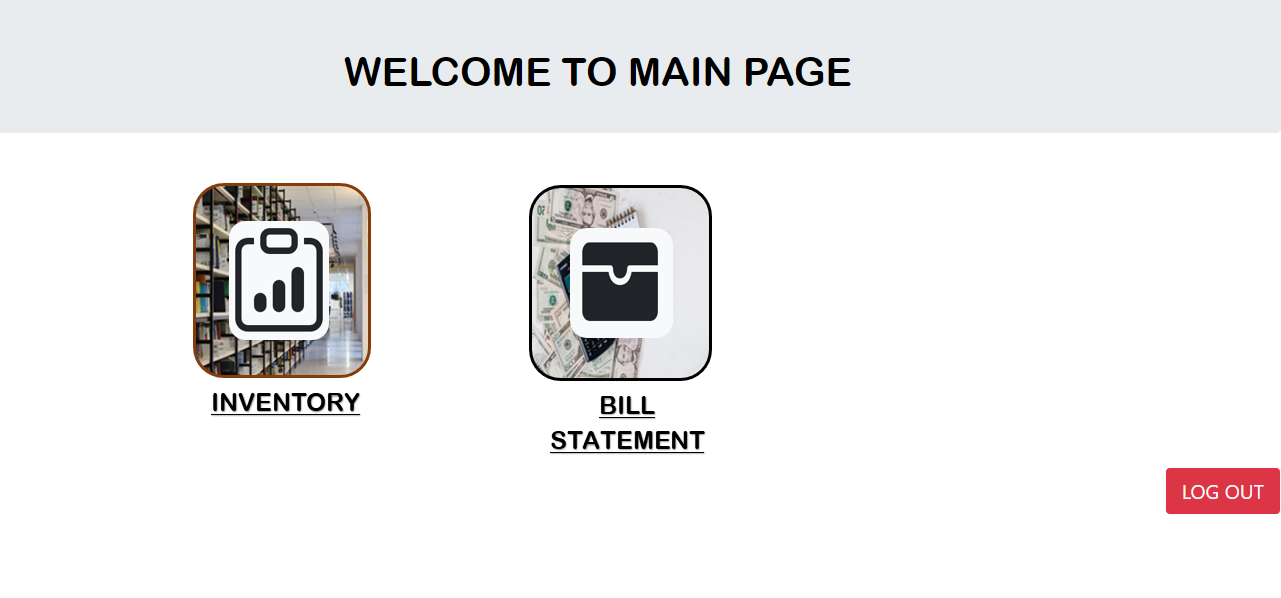
* Centralization of the data access logic.
* Substitution point for the unit tests.
* Flexible architecture that can be adapted as the overall design of the application evolves

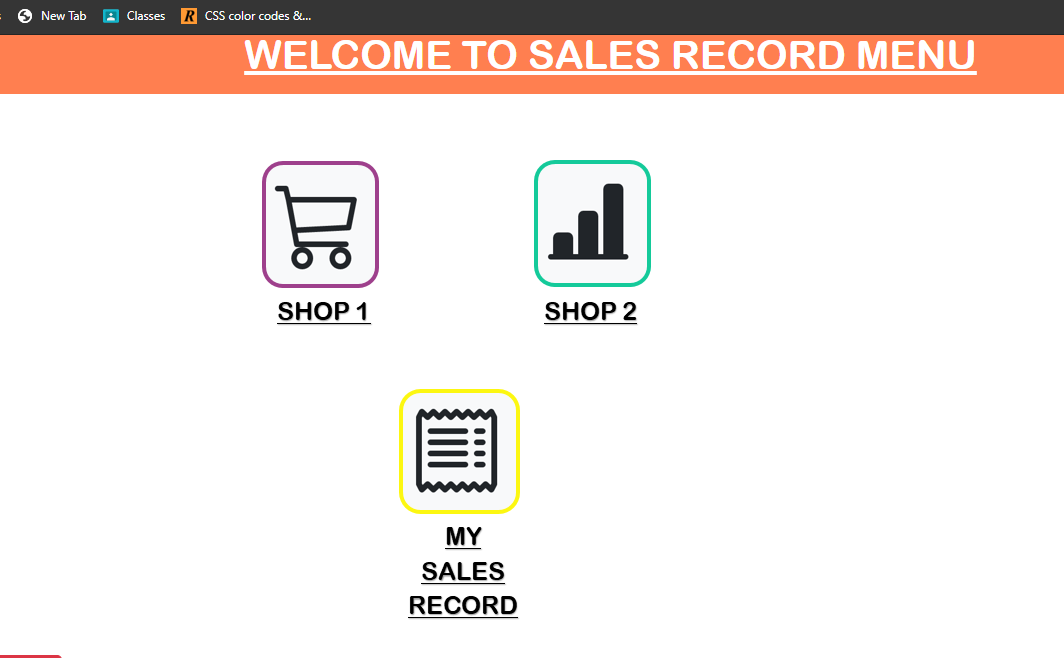


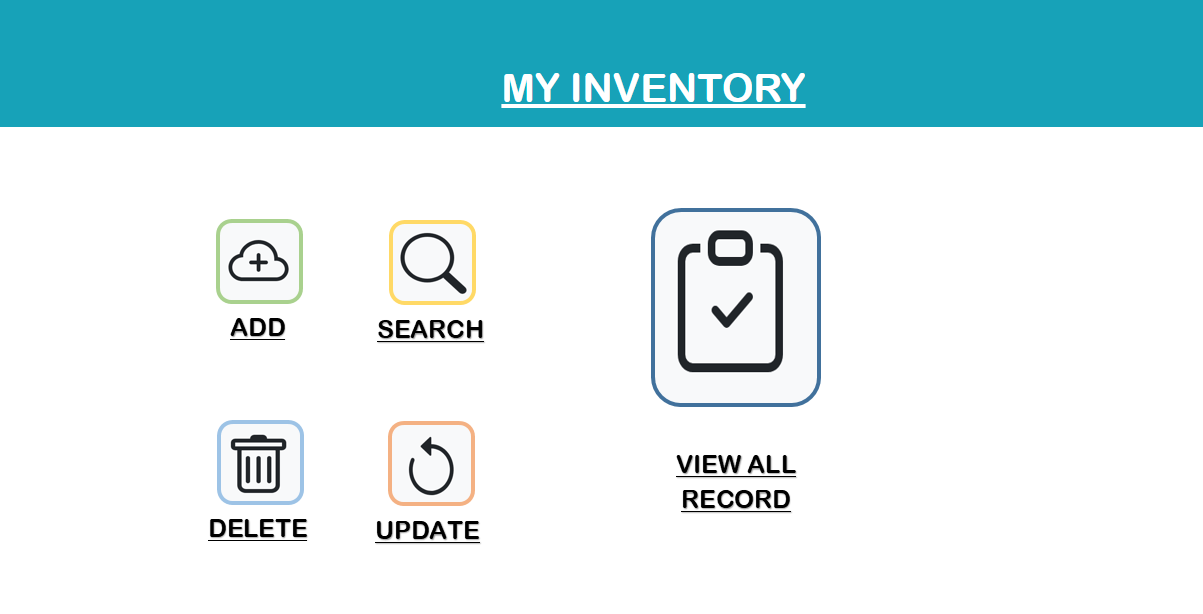
**User Interface**

The user interface design was very thoroughly considered since the success of this system relies very heavily on user acceptance. As mentioned previously, most users of this system will be Inexperienced and thus a simplistic interface is mandatory. As such, we decided that each screen should present the user with as little information and as few choices as possible. In this manner, the users will not be inundated with data; they will simply be presented with a small number of choices that they can select from in order to get the information they need.

The first screen the users will be presented with is a login and password screen. All employees are issued logins and passwords so they can access the. Users can use this information to log onto the system and then change their passwords once they are inside. This takes care of the security requirements. Once they have successfully entered the system, users will be presented with a screen composed of multiple “tabs” which look similar to the tabs where labels are put on paper files. These tabs bear various labels, clicking on any of them brings up a different screen with information appropriate to that subject. Our list of functional requirements specifies a list of data we are required to be able to accept from the user, or display to the user. The following is a tree of screens and functions that our interface will incorporate in order to facilitate this input/output (I/O) of information. Items appearing in boxes, written in regular print are screens that a user can navigate to i.e. there will be a ‘Login’ screen, and a ‘Main Options’ screen.







**Conclusion**

CONCLUSIONS

In the light of software engineering methods, we gather re-

quirements, analyze and figure out the workflow, design me-

thods and structures, construct scenarios, make tests, code

the software, debug the faults and bugs, and finally we obtain

a new inventory management system software. Our first aim

is to develop the IMS software that is able

rements gathered. During the development process, there

are many feedbacks, leaded us to re-design.

The software has the ability to track, to keep history, to

give detailed reporting for each inventory. Also modules of

the software manage the services and operations for invento-

ries and users. Briefly, we could satisfy the user’s require-

ments. It is also a scalable and a flexible solution. No matter

how fast bu

ent system software can adapt to meet or exceed the re-

quirements.

A new, functionally content rich software architecture

model of an inventory management system is developed, pre-

sented, implemented and discussed. This le

monly appear

believe this is a significant contribution.

In the light of software engineering methods, we gather requirements, analyze and figure out the workflow, design methods and structures, construct scenarios, make tests, code the software, debug the faults and bugs, and finally we obtain a new inventory management system software. Our first aim is to develop the IMS software that is able to meet the requirements gathered. During the development process, there are many feedbacks, leaded us to re-design. The software has the ability to track, to keep history, to give detailed reporting for each inventory. Also modules of the software manage the services and operations for inventories and users. Briefly, we could satisfy the user’s requirements. It is also a scalable and a flexible solution. No matter how fast businesses growing, proposed inventory management system software can adapt to meet or exceed the requirements. A new, functionally content rich software architecture model of an inventory management system is developed, presented, implemented and discussed.

software should be able to support at least three simultaneous users.  95% of the transactions shall be processed in less than one second.  Data should be secured and backed up every quarter hour.  Power supply should have a back up and a disaster recovery plan.  System should be operable 24 hours a day and accessible in real-time.  Encryption will Enable SecurityINTRODUCTION

Inventory Management is one of the basic problems for a

company. It may cause a lot of paperwork, if there is no au-

tomated system available. Implementing such a system is

possible but there are a lot of preliminary works such as de-

termination of the requirements, system structure decision –

software requirements, barcode system selection and deter-

mination of the software tools

INTRODUCTION

Inventory Management is one of the basic problems for a

company. It may cause a lot of paperwork, if there is no au-

tomated system available. Implementing such a system is

possible but there are a lot of preliminary works such as de-

termination of the requirements, system structure decision –

software requirements, barcode system selection and deter-

mination of the software too