

Министерство образования Республики Беларусь  
Учреждение образования  
Белорусский государственный университет  
Информатики и радиоэлектроники  
Кафедра инженерной психологии и эргономики

На правах рукописи

УДК

Силинский  
Ромуальд Андреевич

ЭРГОНОМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

**АВТОРЕФЕРАТ**

на соискание академической степени  
магистра технических наук

1-23 80 08 Психология труда, инженерная психология, эргономика

Магистрант Р.А. Силинский

Научный руководитель  
И.Г. Шупейко, кандидат  
психологических наук, доцент

Заведующий кафедрой ИПиЭ  
К.Д. Яшин, кандидат технических  
наук, доцент

Нормоконтролер  
Е.С. Иванова,  
ассистент кафедры ИПиЭ

Минск 2016

## ВВЕДЕНИЕ

Постоянный и ускоряющийся рост информации привел к существенному усложнению анализа имеющихся данных. Для аналитика просмотреть весь этот объем информации без специализированной помощи не представляется возможным. Как следствие, данные часто используются не эффективно или вовсе не используются, что приводит к издержкам, которые мешают дальнейшему развитию предприятия.

Поэтому одной из главных задач любого предприятия является совершенствование методов анализа данных, изучение которых помогает принять решения по поводу любых проблем данного бизнеса, в частности, анализ и учет имеющейся продукции. По полученному анализу можно строить политику предприятия, корректировать объемы и направленность продаж, проводить обусловленный выбор поставщиков продукции и т.п.

Для учета имеющейся на предприятии продукции используются различные программные средства. Однако анализ рынка, существующего программного обеспечения в области анализа данных показал, что подавляющая масса программ не удовлетворяет в полном объеме потребностям организаций. И стоимость таких программных средств достаточно велика.

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Целью работы является разработка автоматизированной системы с показателем эргономичности не менее 0,7.

Задачи на исследование:

1. Провести анализ научно-технической литературы и имеющихся аналогов систем учета товаров.
2. Выполнить эргономическое проектирование автоматизированной системы учета товаров.
3. Осуществить программную реализацию системы учета товаров.

Объектом исследования в данной работе является информационная система учета товаров.

Предметом исследования является показатель эргономичности системы

## СОДЕРЖАНИЕ РАБОТЫ

При создании современных систем ученым необходимо опираться на множество факторов: стоимость системы, её быстродействие, удобство пользования ею, безопасность функционирования и др. Эти факторы можно отнести к одной из трёх групп: требования к человеку, требования к оборудованию и требования к рабочей среде. Идея эргономического проектирования системы состоит в том, чтобы рассматривать человека, технику и условия среды как звенья сложного функционирующего объекта, определить которые без взаимосвязи не представляется возможным.

Проектируемая система должна обеспечивать выполнение следующих функций: отображение сохраненной базы данных; создание новой базы данных; добавление нового элемента; отменить добавление нового элемента; выделение добавленного элемента; выделение необходимого элемента базы данных; удаление элемента базы данных; редактирование элемента базы данных; поиск необходимого элемента в базе данных; поиск элемента в отдельном столбце; очистить поле поиска; упорядочить элементы по убыванию; упорядочить элементы по возрастанию; осуществить возврат в начало базы данных; отображение номер текущей страницы базы данных; переход на определенную страницу базы данных по номеру; переход на следующую страницу базы данных; переход к концу базы данных; выбор количества отображаемых элементов на страницу; отображение всплывающих подсказок об кнопках (иконках); отображение диалогового окна с подтверждением выхода из базы данных; выход из базы данных.

Содержание функций системы можно описать следующим образом:

1. Отображение сохраненной базы данных осуществляется путем открытия ее с помощью ярлыка с использованием мыши или клавиатуры и монитора компьютера.
2. Создание новой базы данных осуществляется путем нажатия определенной кнопки в системе, расположенной рядом с названием открытой базы данных, с использованием мыши компьютера.
3. Добавление нового элемента осуществляется путем нажатия определенной кнопки в системе с использованием мыши компьютера.
4. Отмена добавление нового элемента осуществляется путем нажатия определенной кнопки в системе с использованием мыши компьютера.
5. Выделение добавленного элемента осуществляется системой автоматически после завершения добавления элемента.

6. Выделение необходимого элемента базы данных осуществляется с помощью щелчка по нему левой кнопкой мыши компьютера.

7. Удаление элемента базы данных осуществляется с помощью выбора необходимого элемента щелчка по нему левой кнопкой мыши и нажатием кнопки системы, отвечающей за удаление.

8. Редактирование элемента базы данных с помощью выбора необходимого элемента щелчка по нему левой кнопкой мыши и нажатием кнопки системы, отвечающей за редактирование.

9. Поиск необходимого элемента в базе данных осуществляется путем открытия строки поиска нажатием на специальную кнопку с помощью мыши компьютера и ввода необходимого в открывшееся поле с помощью клавиатуры компьютера.

10. Поиск элемента в отдельном столбце осуществляется путем ввода необходимого в поисковое поле нужного столбца с помощью клавиатуры компьютера.

11. Очистить поле поиска можно с помощью нажатия соответствующей иконки-крестика левой кнопкой мыши или удалением всех символов поля с помощью клавиатуры компьютера.

12. Упорядочить элементы по убыванию можно с помощью нажатия левой кнопки мыши компьютера на соответствующую иконку-стрелку, расположенную рядом с полем-заголовком столбца.

13. Упорядочить элементы по возрастанию можно с помощью нажатия левой кнопки мыши компьютера на соответствующую иконку-стрелку, расположенную рядом с полем-заголовком столбца.

14. Осуществить возврат в начало базы данных можно с помощью нажатия левой кнопки мыши компьютера на соответствующую иконку-стрелку, расположенную напротив поля-заголовка базы данных.

15. Отображение номер текущей страницы базы данных осуществляется системой автоматически и выводится на экран монитора.

16. Переход на определенную страницу базы данных осуществляется путем ввода необходимого номера страницы с помощью клавиатуры компьютера в специальное поле, с отображением текущего номера.

17. Переход на следующую страницу базы данных осуществляется путем нажатия на кнопку переход страницы с помощью мыши компьютера.

18. Переход к концу базы данных осуществляется путем нажатия на кнопку перехода с помощью мыши компьютера.

19. Выбор количества отображаемых элементов на страницу (20, 40 или 60) осуществляется путем выбора соответствующего числа в выпадающем меню с помощью мыши компьютера.

20. Отображение всплывающих подсказок об кнопках осуществляется системой при наведение курсора мыши компьютера на интересующий элемент.

21. Отображение диалогового окна с подтверждением выхода из базы данных осуществляется системой при выполнении выхода из базы данных.

22. Выход из базы данных осуществляется путем закрытия вкладки в браузере или же остановкой развернутого локального сервера.

Эргономическая оценка инженерных решений – это комплекс научно-технических и организационно-методических мероприятий по оценке выполнения в проектных документах и в образцах СЧМ эргономических требований технического задания, нормативно-технических и руководящих документов, а также разработка рекомендаций для устранения отступлений от этих требований. Указанная оценка проводится при обосновании выполнения каждого этапа опытно-конструкторской разработки: технического предложения, эскизного проекта, рабочего проекта.

Исходными материалами для эргономической оценки служат техническое задание на разработку систем, техническая документация, показывающая результаты эргономического проектирования, конструкторские документы, образцы системы «человек-машина – среда» и их составные части.

На практике эргономическая оценка представляет собой определение соответствия показателей объекта оценки эргономическим требованиям (ЭТ) и установление эргономического уровня качества оцениваемого объекта, т. е. степени реализации эргономических требований.

Для проведения эргономической оценки пользовательского интерфейса проектируемой системы будем использовать экспертный метод, при этом в качестве эксперта выступает сам автор проекта.

Проводим оценку значений единичных эргономических показателей. При этом рекомендуемые значения единичных эргономических показателей устанавливаются на основе действующих нормативно-технических документов и эргономических справочников.

Единичные эргономические показатели оцениваются по бинарной шкале, они принимают значение, равное "1", если фактическое значение показателя соответствует рекомендуемому, и равное "0", если оно ему не соответствует.

Групповой эргономический показатель ЭП<sub>гр</sub> рассчитывается как общая оценка по группе единичных показателей по формуле 2.1:

$$\text{ЭП}_{\text{гр}} = \sum 1 / \sum 1 + \sum 0, \quad (2.1)$$

где  $\sum 1$  – суммарное число случаев, когда имеет место соответствие единичных показателей эргономическим требованиям;

$\sum 0$  – суммарное число случаев, когда соответствия нет.

Очевидно, что  $\sum 1 + \sum 0$  - это общее число единичных показателей в группе, поэтому групповой эргономический показатель изменяется в пределах  $0 \leq \text{ЭПгр} \leq 1$ , имеет смысл эмпирической вероятности и служит мерой соответствия характеристик СЧМ эргономическим требованиям данной группы.

Эргономические свойства СЧМ определяются как некоторая совокупность групповых эргономических показателей, при этом чаще всего применяется аддитивная функция:

$$\text{ЭСВ} = \sum \alpha_{ni} * \text{ЭПгр}_j, \quad (2.2)$$

где  $\alpha_{ni}$  – нормированные весовые коэффициенты, сумма которых должна быть равна единице, т.е. ( $\sum \alpha_{ni} = 1$ ).

Для оцениваемого эргономического свойства «управляемость» выбираем величины весовых коэффициентов из таблицы 1:

Таблица 1 – Значения весовых коэффициентов для оценки эргономического свойства «управляемость»

Групповой ЭП	Значение весового коэффициента
Психофизиологический	0,25
Психологический	0,4
Физиологический	0,25
Гигиенический	0,1

С учетом данных таблицы 2 и таблицы 3 по формуле (\*\*\*) определяем количественное значение эргономического свойства «управляемость»

$$\text{ЭСВ} = (0,25 * 1) + (0,4 * 0,72) + (0,25 * 0,9) + (0,1 * 1) = 0,863$$

Следовательно, эргономичность нашей системы равна 0,863.

Разработка программной части. Первое, с чего необходимо начать – описать и обосновать выбор используемых технологий для создания нашего проекта. Начнем мы с шаблона проектирования. Также, при создании данной работы были задействованы: java 7, git (система контроля версий), JBoss ear3

(сервер), mysql 5.5, hibernate (для работы java с базой данных), spring (основной фреймворк для всей архитектуры).

Структура и реализация программного модуля. Среди всех файлов проекта можно выделить следующие типы файлов: Java классы, в которых содержится каркас и логика приложения; HTML страницы; конфигурационные файлы; таблицы стилей пользовательского интерфейса; файлы с JavaScript кодом.

HTML страницы находятся в директории web. Конфигурационные файлы находятся в директории web/WEB-INF.

Среди конфигурационных файлов можно выделить следующие: log4j.properties—содержит настройки библиотеки логирования log4j; содержит настройки, ограничения; jpa-context.xml – настройки слоя доступа к данным (настройка соединения с базой данных, параметры соединения и т.п.); web.xml—основной файл настройки Java веб-приложения. Определяет путь доступа к приложению, его структуру и главный сервлет.

Код дескриптора развёртывания web.xml представлен в приложении. В этом файле описывается: путь к корневому контексту (содержит настройки Spring Security и JPA); путь к файлу, содержащему настройки ресурсов, представлений (JSP), контроллеров; настройки кодировки;

Приложение также имеет внешние зависимости от следующих сторонних библиотек (указаны в файле pom.xml): фреймворк Spring для построения и выполнения приложений Java; Log4j—библиотека логирования; JPA—ORM технология для работы с базой данных; JSTL—библиотека тэгов для jsp страниц; Jackson—библиотека для работы с JSON; Maven3—сборка приложения.

Результатом компиляции приложения является веб-архив (WAR-файл), который может быть развернут на любом контейнере сервлетов, реализующих спецификацию JSP (Tomcat, JBoss, WebLogic и др.).

Реализация слоя данных. JPA—API, входящий с версии Java5 в состав платформ JavaSE и JavaEE, предоставляет возможность сохранять в удобном виде Java-объекты в базе данных.

Существует несколько реализаций этого интерфейса, одна из самых популярных использует для этого Hibernate. JPA реализует концепцию ORM.

ORM—технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». Существуют как проприетарные, так и свободные реализации этой технологии.

В объектно-ориентированном программировании объекты в программе представляют объекты из реального мира. В качестве примера можно рассмотреть адресную книгу, которая содержит список людей с нулём или



более телефонов и нулём или более адресов. В терминах объектно-ориентированного программирования они будут представляться объектами класса «Человек», которые будут содержать следующий список полей: имя, список (или массив) телефонов и список адресов.

Суть задачи состоит в преобразовании таких объектов в форму, в которой они могут быть сохранены в файлах или базах данных, и которые легко могут быть извлечены в последующем, с сохранением свойств объектов и отношений между ними. Эти объекты называют «хранимыми». Исторически существует несколько подходов к решению этой задачи.

Entity (Сущность) – POJO-класс, связанный с БД с помощью аннотации @Entity или через XML. К такому классу предъявляются следующие требования: должен иметь пустой конструктор (public или protected); не может быть вложенным, интерфейсом или enum; не может быть final и не может содержать final-полей/свойств; должен содержать хотя бы одно @Id-поле.

При этом entity может: содержать непустые конструкторы; наследоваться и быть наследованным; содержать другие методы и реализовывать интерфейсы.

Entities могут быть связаны друг с другом (один-к-одному, один-ко-многим, многие-к-одному и многие-ко-многим).

Взаимодействие представления и модели. В разрабатываемом приложении модель и представление взаимодействует посредством контроллера. Контроллеры Spring MVC выполняют следующие задачи: обработка полученных от сервисов данных для передачи на слой отображения по запросу; принятие данных со слоя взаимодействия с пользователями, их предварительная обработка, проверка и передача слою сервисов для дальнейшей обработки.

Каждый контроллер представляет собой java-класс, помеченный аннотацией @Controller. Контроллер связан лишь с одной страницей, которую он обрабатывает. Связь между контроллерами осуществляется программно, с помощью передачи параметров между ними.

Реализация пользовательского интерфейса. Пользователь взаимодействует с системой с помощью веб-браузера. Взаимодействия сервера приложения с веб-браузером пользователя происходит по протоколу HTTP с помощью языка разметки HTML. Язык HTML интерпретируется веб-браузером и отображается в виде документа, в удобной для человека форме. Внешний вид системы определяется дизайном и HTML-версткой.

## ЗАКЛЮЧЕНИЕ

Изучены самые популярные системы по учету комплектующих и были выявлены их недостатки. Главным является неудобный, неприспособленный к пользователю интерфейс. В ходе реализации продукта выявлено, что одним из самых удобных способов

Проведено эргономическое проектирование системы для исследования деятельности оператора. В результате получено описание автоматизированной системы учета товаров, проведено обоснование эргономических требований к системе, определены и раскрыты функции, выполняемые каждым из звеньев системы, разработаны структура системы, алгоритмы работы пользователей, сценарий информационного взаимодействия человека и персонального компьютера.

Разработана автоматизированная система учета товаров, которая позволяет легко и удобно вести учет необходимых комплектующих, осуществляет сортировку и фильтрацию элементов таблицы. В дополнение к системе спроектированы и вспомогательные функции подсчета статистики, которые будут полезны в процессе работы. Таким образом, был получен готовый программный продукт, который пригоден к использованию на предприятии.