

Software Requirements Specification  
For  
**Healthcare Managenment System**

Version 1.0 Approved

**Prepared by: Ayush Singh**

**HCL Training**

Date: 13/01/2026

Submitted in partial fulfillment  
Of the requirements of  
Software Engineering(Agile Methodology)

# Table of Contents

1. **Abstract**
2. **Introduction**
  - 2.1 Introduction
  - 2.2 Problem Identification
  - 2.3 Need of the Project
  - 2.4 Project Scheduling
  - 2.5 Objectives
3. **Software Requirement Specification (SRS)**
  - 3.1 Purpose
  - 3.2 Scope
  - 3.3 Hardware Requirement / Software Requirement
  - 3.4 Tools
  - 3.5 Software Process Model
4. **System Design**
  - 4.1 Data Dictionary
  - 4.2 ER Diagram
  - 4.3 Data Flow Diagram (DFD)
  - 4.4 System Flow Chart / Object Diagram / Class Diagram / State Diagram / Activity Diagram / Use Case Diagram
5. **Implementation**
  - 5.1 Program Code
  - 5.2 Output Screens
6. **Testing**
  - 6.1 Test Data
  - 6.2 Test Result
7. **User Manual**
  - 7.1 How to Use Project Guidelines
  - 7.2 Screen Layouts and Description
8. **Project Applications and Limitations**
9. **Conclusion and Future Enhancement**
10. **Bibliography and References**

## **1. Abstract**

The Healthcare Management System is a software application designed to automate and manage healthcare-related activities such as patient registration, appointment scheduling, medical records management, billing, and reporting. In traditional healthcare systems, most processes are handled manually, leading to inefficiency, data inconsistency, delays, and higher chances of human error. With the rapid advancement of information technology, there is a strong need for a digital healthcare solution that ensures accuracy, efficiency, and improved patient care.

This project aims to develop a Healthcare Management System using the Agile software development methodology. The system provides a centralized platform where patients, doctors, and administrators can interact efficiently. Patients can register, book appointments, and view medical history. Doctors can manage appointments, update patient records, and prescribe treatments. Administrators can manage users, departments, and system data securely. Agile methodology is used to develop the system in iterative sprints, allowing continuous feedback, flexibility, and timely delivery of functional modules.

The system focuses on data security, reliability, and user-friendliness. By digitizing healthcare processes, the system reduces paperwork, improves data accuracy, and enhances overall healthcare service quality. This project demonstrates how Agile practices help in building scalable, maintainable, and efficient healthcare software solutions. Future enhancements can include telemedicine, mobile app integration, and AI-based diagnosis support.

## **2. Introduction**

The Healthcare Management System is a software solution designed to streamline and automate the management of healthcare operations in hospitals, clinics, and medical centers. Traditional healthcare systems rely heavily on manual processes for patient registration, appointment scheduling, medical record management, and billing, which often leads to errors, delays, and inefficiencies. By implementing a digital system,

healthcare providers can improve service delivery, ensure data accuracy, and enhance patient satisfaction.

## 2.1 Introduction

Healthcare organizations require a reliable system to manage patient information, track appointments, maintain medical histories, and generate billing reports efficiently. The Healthcare Management System provides a centralized platform for patients, doctors, and administrators to interact seamlessly. Patients can register, book appointments, and view medical records, while doctors can manage patient consultations and treatment details. Administrators can manage departments, staff, and system data securely.

## 2.2 Problem Identification

Some of the key problems in traditional healthcare management include:

- Manual patient registration and record-keeping
- Delays in appointment scheduling and follow-ups
- Difficulty in accessing and updating medical records
- Increased chances of human error in patient data and billing
- Inefficient communication between patients, doctors, and administrative staff

## 2.3 Need of the Project

The need for a Healthcare Management System arises from the necessity to:

- Automate healthcare processes and reduce manual workload
- Ensure accurate storage and retrieval of patient medical records
- Enable quick appointment booking and management
- Improve communication between patients, doctors, and staff
- Enhance overall healthcare service quality and efficiency

## 2.4 Project Scheduling

The project is developed using the **Agile methodology**, which focuses on iterative development and continuous improvement. The scheduling of the project is planned in multiple **sprints**, each delivering functional modules of the system:

Sprint No.	Sprint Name	Activities	Duration
1	Requirement Analysis & Planning	Requirement gathering, feasibility study, backlog	1 Week
2	System Design	Database design, ER diagram, DFD, UI design	1 Week
3	Patient & Doctor Modules	User registration, login, profile management	1 Week
4	Appointment & Records Management	Appointment booking, medical record management	2 Weeks
5	Testing, Deployment & Documentation	System testing, bug fixing, deployment, manual	1 Week

## 2.5 Objectives

- To develop an automated Healthcare Management System
- To reduce manual workload and paperwork in healthcare operations
- To improve the accuracy and accessibility of patient records
- To facilitate quick and efficient appointment scheduling
- To ensure secure access to patient and doctor data
- To provide administrators with tools to manage hospital operations efficiently
- To implement the system using Agile methodology for iterative development

## 3. Software Requirement Specification (SRS)

The Software Requirement Specification (SRS) document defines the functional and non-functional requirements of the Healthcare Management System. It provides a clear understanding of system expectations, functionality, constraints, and tools required for development. The SRS serves as a guide for developers, testers, and stakeholders to ensure that the system meets its objectives.

### 3.1 Purpose

The purpose of this SRS is to define all the necessary requirements for the Healthcare Management System. It ensures that developers have a clear understanding of the system's functionality and that the final product meets the expectations of healthcare providers and patients. This document also serves as a reference for system testing, deployment, and future maintenance.

The system aims to:

- Automate patient registration and appointment scheduling
- Maintain secure and accurate medical records
- Enable doctors to manage patient consultations and treatment details
- Provide administrators with tools to manage hospital operations efficiently

### 3.2 Scope

The Healthcare Management System provides a centralized platform for managing healthcare operations. Its scope includes:

- **Patient Module:** Registration, login, view medical history, book appointments
- **Doctor Module:** Manage appointments, update patient records, prescribe treatments
- **Administrator Module:** Manage users, departments, and system data
- **Billing Module:** Generate invoices and manage payment records
- **Reporting:** Generate reports on appointments, treatments, and revenue

The system is designed for hospitals, clinics, and healthcare centers. It ensures improved patient care, faster service delivery, and reduced administrative workload. Future enhancements may include telemedicine, mobile apps, and AI-based diagnostics.

### 3.3 Hardware Requirement / Software Requirement (Minimum)

#### Hardware Requirements:

- Processor: Intel i3 or higher
- RAM: Minimum 4 GB
- Hard Disk: At least 20 GB free space
- Monitor: 15-inch or higher
- Internet Connection: Required for online access

#### Software Requirements:

- Operating System: Windows 10 / Linux
- Web Browser: Google Chrome / Mozilla Firefox
- Database: MySQL
- Server: Apache / Tomcat

- Programming Language: Java / Node.js

### 3.4 Tools

The following tools are used for development:

- **Frontend:** HTML, CSS, JavaScript (for web interface)
- **Backend:** Java / Node.js (for server-side processing)
- **Database:** MySQL (for storing patient, doctor, appointment, and billing data)
- **IDE:** Eclipse / Visual Studio Code
- **Version Control:** Git (for source code management)

These tools ensure that the system is scalable, maintainable, and secure.

### 3.5 Software Process Model

The **Agile Software Development Model** is used for this project. Agile methodology allows iterative and incremental development, ensuring that each module is tested and validated in sprints. This approach allows:

- Continuous feedback from users and stakeholders
- Early detection and correction of errors
- Flexible adaptation to changing requirements
- Timely delivery of functional software modules

The project is divided into multiple sprints, each focusing on a specific feature such as patient registration, appointment scheduling, medical records management, billing, and reporting. Agile ensures a high-quality, user-centric, and efficient Healthcare Management System.

## 4.1 Data Dictionary

The Data Dictionary lists all major entities and their attributes used in the system:

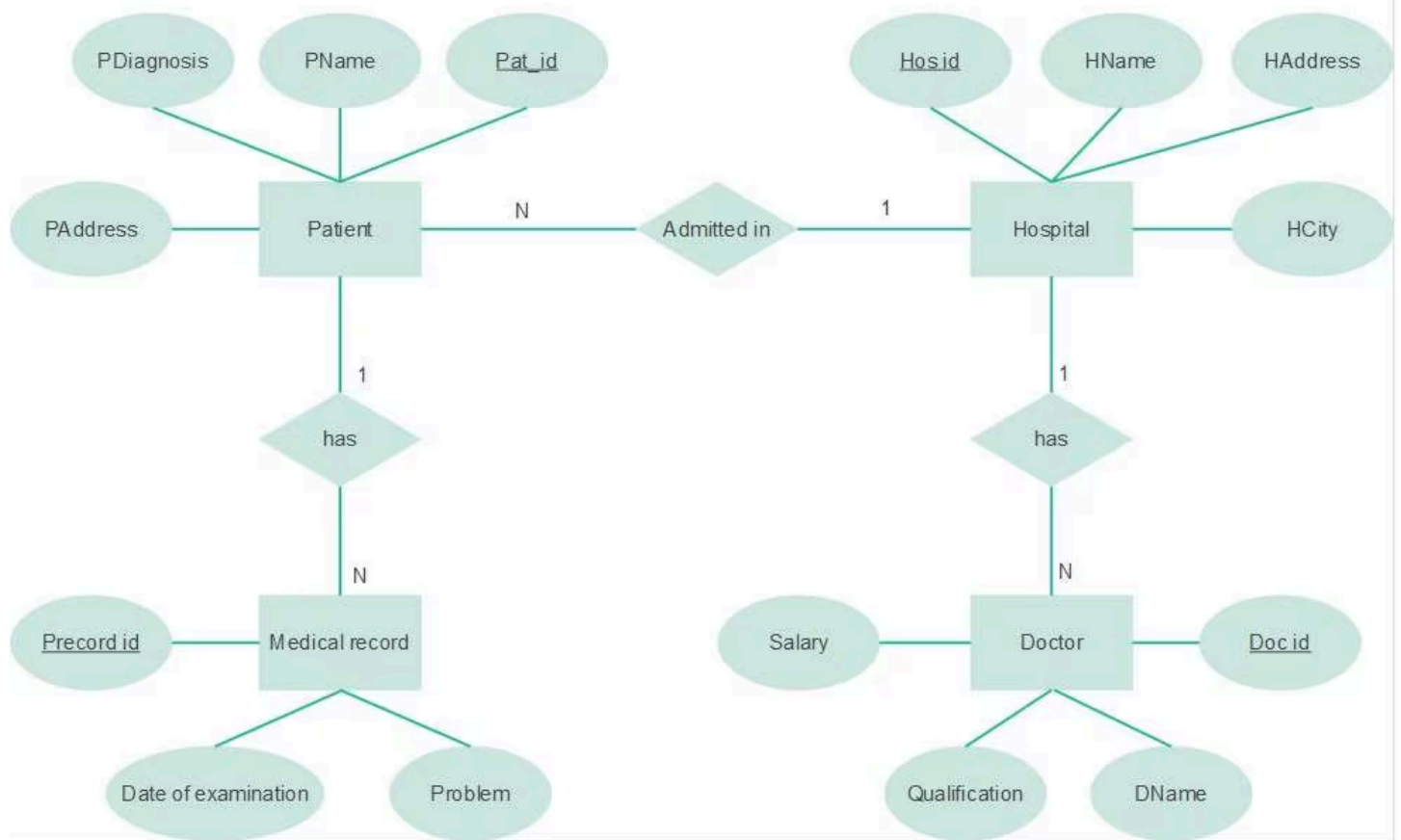
Entity	Attribute	Description
Patient	patient_id	Unique ID of patient
	name	Full name of patient
	age	Age of patient
	gender	Gender of patient
	email	Patient email for login and communication
	phone	Contact number
	doctor_id	Unique ID of doctor
	name	Doctor's full name
	specialization	Field of expertise
	email	Doctor email for communication
	phone	Contact number
	appointment_id	Unique ID of appointment
	patient_id	ID of the patient
	doctor_id	ID of the doctor
	appointment_date	Scheduled date of appointment
	status	Appointment status (Confirmed / Cancelled)
	record_id	Unique ID of medical



		record
	patient_id	ID of the patient
	doctor_id	ID of doctor creating record
	diagnosis	Diagnosis details
	prescription	Prescribed medicines or treatment
Payment	payment_id	Unique ID of payment
	patient_id	ID of patient
	amount	Payment amount
	payment_mode	Card / UPI / Net Banking
	payment_status	Successful / Failed

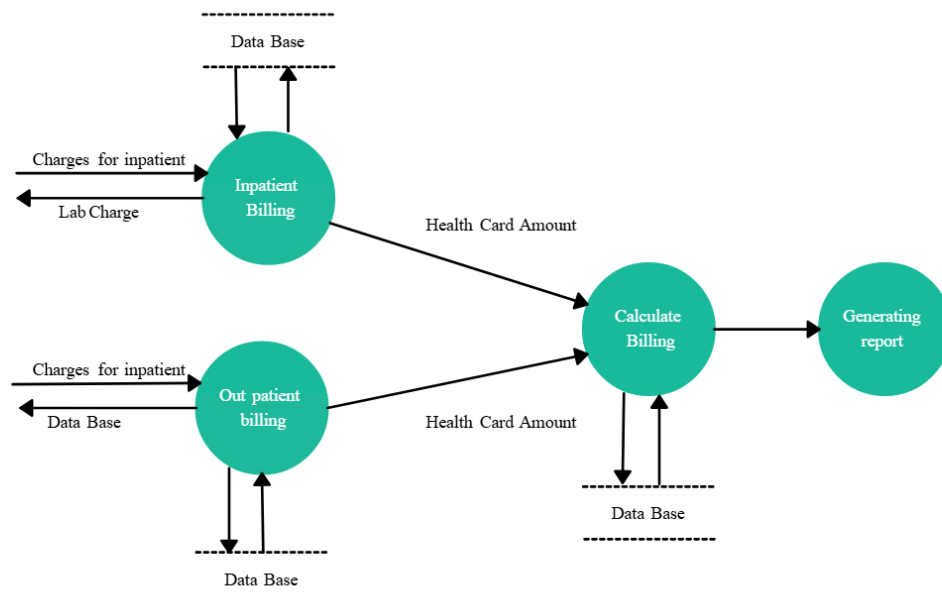
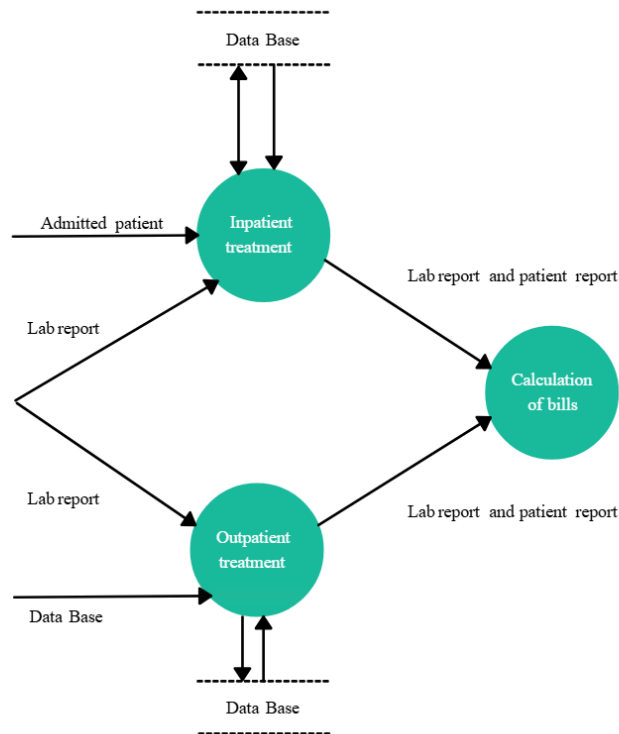
### 4.2 ER Diagram (Entity-Relationship Diagram)

The **ER Diagram** shows relationships between the main entities in the Healthcare Management System:



### 4.3 Data Flow Diagram (DFD)

The **DFD** shows the flow of data between users and the system:



## 5. Implementation

Implementation is the phase where the system design is converted into a working application. In the Healthcare Management System, each module—such as patient registration, doctor management, appointment booking, medical record management, and billing—is developed using Java for backend

processing and HTML/CSS/JavaScript for the frontend interface. The system is implemented in a modular and iterative manner following Agile methodology.

## 5.1 Program Code (Sample)

### Patient Class (Java)

```
public class Patient {  
  
    private int patientId;  
  
    private String name;  
  
    private int age;  
  
    private String gender;  
  
    private String email;  
  
  
    public Patient(int patientId, String name, int age, String gender, String email) {  
  
        this.patientId = patientId;  
  
        this.name = name;  
  
        this.age = age;  
  
        this.gender = gender;  
  
        this.email = email;  
  
    }  
  
    public void displayPatientDetails() {  
  
        System.out.println("Patient ID: " + patientId);  
  
        System.out.println("Name: " + name);  
  
        System.out.println("Age: " + age);  
  
        System.out.println("Gender: " + gender);  
  
        System.out.println("Email: " + email);  
  
    }  
  
    public void bookAppointment(int doctorId, String date) {
```

```
        System.out.println("Appointment booked with Doctor ID " + doctorId + " on " + date);
    }
}
```

### **Appointment Class (Java)**

```
public class Appointment {

    private int appointmentId;

    private int patientId;

    private int doctorId;

    private String appointmentDate;

    private String status;


    public Appointment(int appointmentId, int patientId, int doctorId, String appointmentDate) {

        this.appointmentId = appointmentId;

        this.patientId = patientId;

        this.doctorId = doctorId;

        this.appointmentDate = appointmentDate;

        this.status = "Confirmed";

    }


    public void displayAppointment() {

        System.out.println("Appointment ID: " + appointmentId);

        System.out.println("Patient ID: " + patientId);

        System.out.println("Doctor ID: " + doctorId);

        System.out.println("Date: " + appointmentDate);

        System.out.println("Status: " + status);

    }

}
```

## 5.2 Output Screens

The major screens of the Healthcare Management System include:

### 1. Login Screen

- Input: Email and password
- Function: Secure authentication for patients, doctors, and admin

### 2. Registration Screen

- Input: Patient or doctor details
- Function: Store user information in the database

### 3. Patient Dashboard

- Features: View profile, book appointments, view medical history

### 4. Doctor Dashboard

- Features: View appointments, update medical records, prescribe treatments

### 5. Appointment Booking Screen

- Input: Select doctor, date, and time
- Output: Appointment confirmation

### 6. Payment Screen

- Input: Payment details
- Output: Payment confirmation message

### 7. Admin Dashboard

- Features: Manage users, view appointments, generate reports

## 6. Testing

Testing is a crucial phase of the software development life cycle. It ensures that the Healthcare Management System works correctly, meets all functional requirements, and provides reliable performance. The purpose of testing is to identify errors, verify system functionality, and ensure that the system is user-friendly, secure, and efficient.

Both **functional testing** (checking modules like registration, login, appointments) and **non-functional testing** (performance, security) are performed.

### 6.1 Test Data

The following test data is used to validate different modules of the system:

Test Case ID	Module	Input Data	Expected Output
TC01	User Registration	Name, Email, Password	Registration successful
TC02	User Registration	Missing Email	Error: Email required
TC03	Login	Valid Email & Password	Login successful
TC04	Login	Invalid Email/Password	Error: Invalid credentials
TC05	Appointment Booking	Patient ID, Doctor ID, Date	Appointment confirmed
TC06	Appointment Booking	Invalid Date	Error: Invalid date
TC07	Medical Record	Patient ID, Diagnosis, Prescription	Record saved successfully
TC08	Medical Record	Empty Diagnosis	Error: Diagnosis required
TC09	Payment	Patient ID, Amount, Payment Mode	Payment successful
TC10	Payment	Invalid Card Details	Payment failed

## 6.2 Test Result

After executing all test cases, the observed results are summarized below:

Module	Test Case ID	Result	Remarks
User Registration	TC01, TC02	Pass	Validation working correctly
Login	TC03, TC04	Pass	Authentication successful
Appointment Booking	TC05, TC06	Pass	Appointments correctly booked or error shown
Medical Record	TC07, TC08	Pass	Records updated and validated
Payment	TC09, TC10	Pass	Payment processing working correctly

## 7. User Manual

The User Manual provides detailed instructions for using the Healthcare Management System. It helps patients, doctors, and administrators navigate the system and use its features effectively. The system is designed to be user-friendly, secure, and efficient.

### 7.1 How to Use Project Guidelines

#### Step-by-Step Instructions:

- 1. Open the System**
  - Launch the Healthcare Management System in a web browser.
  - Ensure internet connection is active if hosted online.
- 2. User Registration**
  - New users (patients or doctors) must register by providing necessary details such as name, email, password, and contact information.
  - Click “Register” to submit the details.
- 3. Login**
  - Registered users can log in with their email and password.
  - Admin has separate login credentials.
- 4. Patient Module**
  - Book appointments with available doctors.
  - View personal medical history and appointment details.
- 5. Doctor Module**
  - View scheduled appointments.



- Update patient medical records and prescriptions.

## 6. Admin Module

- Manage patient and doctor profiles.
- View reports and monitor hospital operations.

## 7. Appointment Booking

- Patients select a doctor, date, and time.
- Confirm appointment and receive a booking ID.

## 8. Payment Module

- Patients pay using card, UPI, or net banking.
- Payment confirmation is displayed upon successful transaction.

## 9. Logout

- Users should log out after completing their tasks to maintain security.

## 7.2 Screen Layouts and Description

Screen	Description
Login Screen	Input fields for email and password. Buttons for login and new user registration.
Registration Screen	Fields to enter user information: name, email, password, contact number.
Patient Dashboard	Displays user profile, appointments, medical history, and booking options.
Doctor Dashboard	Displays scheduled appointments, patient records, and update options.
Appointment Booking	Allows selection of doctor, date, and time. Displays confirmation upon booking.
Medical Records Screen	Doctors update diagnosis and prescription. Patients can view history.
Payment Screen	Input payment details and select payment mode. Displays payment confirmation.
Admin Dashboard	Manage patients, doctors, appointments, and generate reports.

## 8. Project Applications and Limitations

## Applications

The Healthcare Management System has wide-ranging applications in the medical and healthcare field, such as:

- **Hospitals and Clinics:** Streamlines patient registration, appointment scheduling, and medical record management.
- **Diagnostic Centers:** Manages patient data, test results, and reporting efficiently.
- **Corporate Healthcare:** Supports employee health management and wellness programs.
- **Individual Users:** Patients can view their medical history, book appointments, and make payments online.
- **Small Healthcare Providers:** Helps digitalize records, reduce manual workload, and improve service quality.

The system improves efficiency, reduces errors, and enhances communication between patients, doctors, and administrative staff.

## Limitations

Despite its advantages, the system has certain limitations:

- Requires a stable internet connection for online access.
- Limited to the features implemented in the basic version (e.g., no telemedicine integration).
- Performance may be affected under extremely high user load without server optimization.
- Advanced AI-based diagnostics and automated reminders are not included in the current version.
- Dependent on accurate input; incorrect data may affect output and reports.

# 9. Conclusion and Future Enhancement

## Conclusion

The Healthcare Management System provides a digital platform for managing healthcare operations efficiently. It automates patient registration, appointment scheduling, medical records management, and billing processes. By following the Agile methodology, the system was developed in iterative sprints, allowing continuous feedback, error correction, and early delivery of functional modules.

The system reduces manual work, increases data accuracy, improves communication between patients, doctors, and administrators, and enhances overall healthcare service quality. Testing confirms that the system is reliable, secure, and user-friendly, making it ready for deployment in real-world healthcare scenarios.

## Future Enhancement

The system can be improved in future versions with features like:

- Integration of **telemedicine** for online doctor consultations.
- Mobile application support for Android and iOS devices.
- AI-based medical diagnosis and prescription recommendations.
- Automated reminders via email or SMS for appointments and follow-ups.

- Multi-language support for broader accessibility.
- Advanced reporting and analytics for hospital management.

## 10. Bibliography & References

1. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 8th Edition.
2. IEEE Standard for Software Requirements Specification (SRS).
3. Agile Alliance – *Agile Methodology Guide*, <https://www.agilealliance.org>
4. Oracle Java Documentation, <https://docs.oracle.com/javase/>
5. MySQL Official Documentation, <https://dev.mysql.com/doc/>
6. Sommerville, I., *Software Engineering*, 10th Edition, Pearson Education.