

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

Ex No: 11	SOFTWARE REQUIREMENTS SPECIFICATION A Prototype model - Develop the prototype SRS of the product
Date: 18/10/2021	
Video Link:	https://drive.google.com/file/d/10Pk6mFPuH_s2eYAE9qwrnyAsv-PkGr9/view?usp=sharing

OBJECTIVE

The objective of the experiment is to generate a detailed SRS for the project **AUDITOR MANAGEMENT SYSTEM**.

1. ARRANGEMENT OF CONTENTS

The sequence in which the project report material should be arranged and bound is as follows:

1. Cover Page
2. First Page-same as cover page
3. Table of Contents
4. Chapter 1 – Introduction
5. Chapter 2 – Overall Description
6. Chapter 3 – Specific Requirements
7. Chapter 4 – Technologies used
8. Chapter 5 – Limitation of the project

2. SRS SPECIFICATIONS

- The dimension of the pages should be A4 size.
- Total number of pages for the SRS should be **minimum 40 pages** and should not exceed 50 pages including first page and table of contents page

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

3. PREPARATION FORMAT

- **Cover Page and First Page-** A specimen copy of the “Cover page” and “Title page” is included at page no.4
- **Table of Contents** -A specimen copy of the “Table of Contents” is included at page no.5 and 6
- **Sections** –The SRS may be broadly divided into 5 sections.

1. Introduction

1.1 Purpose – should mention the objective of the project

1.2 Scope – should describe the scope of the project

1.3 Definitions, Acronyms, and Abbreviations – should provide a very brief explanation of the abbreviations and acronyms used

1.4 References – should specify the references taken (includes book, web references)

1.5 Technologies to be used – should specify the technologies to be used (computer languages, IDEs, or any other specific tool)

1.6 Overview – should describe the sections of the SRS

2. Overall Description

2.1 Product Perspective – should mention the interactions

2.2 Software Interface – should specify the front end and back end of the system

2.3 Hardware Interface – should specify the client and server requirements

2.4 Product Functionalities – should specify the functionalities of the system

2.5 User Characteristics – the characteristics of the user should be mentioned

2.6 Constraints – the constraints of the system should be mentioned

2.7 Architecture Diagram – the interaction between the components/modules should be shown

2.8 Use Case Model Description – use case diagram should be drawn

2.9 Class Diagram – class diagram should be drawn (starUML)

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

2.9.1 Sequence Diagrams – sequence diagrams should be drawn

2.10 Database Design – ER and Database schema diagrams should be drawn (starUML)

2.10.1 ER Diagram

2.10.2 Schema

2.11 Assumptions and Dependencies – should specify the assumptions made

3. Specific Requirements

3.1 Use Case Reports – should specify the functionalities of the users

3.2 Supplementary Requirements – specify the non-functional requirements

4. Technologies used

4.1 Explanation of computer languages used

4.2 Explanation of IDEs with its features, plug-ins, etc.,

4.3 Explanation of APIs/Frameworks

4.4 Explanation of developer's documentation on how to use the tools

5. Limitation of the project

SRS PREPARATION GUIDELINES

- Tables and figures in a chapter should be placed in the immediate vicinity of the reference where they are cited.
- Figures should have captions and should be numbered. Eg. Fig 5.8 Variation of speed with load (font size-10,title case).
- Figures, Equations, Tables to be continuously numbered within a chapter eg. Fig 2.1, Fig 2.2...../Table 2.1, Table 2.2.....
- For Figures, graphs, charts, flow diagrams etc., captions should be at the bottom.
- For tables, captions should be above the tabular column.
- Tables and figures should conform to the margin specifications. Large size figures should be reduced to the appropriate size before insertion.

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

- MS Word Times New Roman font, size 12 for body of text; size 13 for sub section titles and bold & CAPS, for sub sections of sub sections, font size is 13, Title Case; size 10 for table heading and figures and captions (title case).
- For chapter title, font size as 14 and bold & CAPS and font Times New Roman.
- Enough matter (text, figures etc.) should be there in a page; avoid white spaces.
- All pages should be numbered continuously; printed at bottom and centre of each page starting from chapter 1
- Exactly 1.5 line spacing.
- Margin: Left=1¼" (1.25), Right=¾" (0.75), top=1", bottom=1".
- Subscripts and superscripts should be properly printed.
- Reference in body to be given with square bracket. **E.g.:** Visual cryptography is a new technique which provides information security which uses simple algorithm unlike the complex, computationally intensive algorithms. [25]

RESULT:

A software requirement specification for modelling a prototype of **AUDITOR MANAGEMENT SYSTEM** was carried out successfully.

AUDITOR MANAGEMENT SYSTEM

SOFTWARE REQUIREMENT SPECIFICATION

[BATCH NO: 03]

Submitted by:

RUBAN GINO SINGH.A - Reg. No. UR20CS2001

In partial fulfillment of the requirements for the
Software Engineering

Under the guidance of

Mr. TITUS



KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES

Declared as Deemed to be University under sec.3 of the UGC Act, 1956

Karunya Nagar, Coimbatore -641 114

OCTOBER 2021

TABLE OF CONTENTS

	Page No
1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, Acronyms and Abbreviations	2
1.4 References	3
1.5 Tools	3
1.6 Overview	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Software Interface	4
2.3 Hardware Interface	4
2.4 Product Functionalities	5
2.5 User Characteristics	5
2.6 Constraints	5
2.7 Architecture Diagram	6
2.8 Use Case Model Description	7
2.9 Class Diagram	8
2.9.1 Sequence Diagrams	9
2.10 Database Design	10
2.10.1 ER Diagram	11
2.10.2 Activity	12
2.11 Assumptions and Dependencies	12
3. Specific Requirements	14
3.1 Use Case Reports	14
3.2 Supplementary Requirements	14
4. Technologies used	14

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

4.1 Languages used	15
4.2 IDE/Tools used	15
4.3 API/Framework documentation	15
4.4 Developer's documentation	15
5. Limitation of the project	16

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

1.Introduction:

Auditor Management System - provides a user-friendly way of managing Auditors in a company. It allows tracking the reports, predicts risks, provides opinions, risk management, and gets the history of the projects from the central repository.

1.1 Purpose:

An auditor management system is to manage the auditors in a particular company. It helps companies streamline their audit processes and comply with regulations or internal policies. Financial records of a company have perfectly verified after every transaction, so to attain this, every company needs auditors to make the monthly records and want to clear tax norms. Audit management helps simplify and well-organize the workflow and collaboration process of compiling audits. Most audit teams heavily rely on email and shared drive for sharing information. This Auditor management system will help the Auditors to fulfil their daily audit needs. Typical tasks such as submitting client requests, sending reminders, and following up on findings had done utilizing broad tools. Investing in the right software could help save time, reduce errors and save on resources. The purpose of a management audit is not to appraise individual executive performance but to evaluate the management team in its effectiveness to work in the interests of shareholders, maintain good relations with employees, and uphold reputational standards. It is important to stress that the management audit assesses the overall management of the company, not the performance of individual managers. This system will assist the auditors and the companies to finish and turned the work at the correct time.

1.2 Scope:

- Create different users like Auditors, managers, and the Board of Directors to assign and receive the privileges and duties.
- All the users need to authenticate to avail of the services from this management system.
- Details of auditors can view easily using the unique id.
- Allows Administrator to add, view, edit and delete auditors records efficiently. And also, to generate reports.

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

- Allows tracking of audit reports can generated in pdf or other usable
- Maintain an FAQ section as guidance for all the users.
- Secure Socket Layer (SSL) connectivity to ensure security for the Auditor management system as a website
- Usage of jQuery for web JS
- Usage of SOA for more effortless connectivity and enhancement

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

- **HTML:** Hypertext Markup Language is used to create static websites
- **CSS:** Cascading Style Sheet used to style the static webpage
- **JS:** JavaScript is a scripting language used for client-side web development
- **Node JS:** A JavaScript library used for backend web development
- **JQUERY:** jQuery is a JavaScript library that emphasizes interaction between JavaScript and HTML.
- **HTTP:** Hypertext Transfer Protocol is a transaction-oriented client/server protocol between the web browser and a web server
- **HTTPS:** Secure Hypertext Transfer Protocol is HTTP over SSL (Secure Socket Layer)
- **SOA:** Service-Oriented Architecture
- **EJB:** Enterprise Java Beans is an architecture for the development and deployment of distributed object application-based server-side software components
- **DB2:** IBM Database 2 is a database management system that provides a flexible and efficient database platform to raise strong "on demand" business applications
- **AJAX:** Asynchronous JavaScript and XML is a technique used in JavaScript to create dynamic web pages
- **WASCE:** Web Sphere Application Server Community Edition is an application server that runs and supports the web applications

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

1.4 REFERENCES

- IBM Database management: <https://www.ibm.com/in-en/products/db2-database>
- Wikipedia: www.wikipedia.com

1.5 TOOLS AND TECHNOLOGIES

- **HTML:** Hypertext Markup Language
- **CSS:** Cascading Style Sheets
- **JS:** JavaScript
- **JQ:** jQuery
- **XML:** Extensible Markup Language
- **UML:** Unified Modelling Language
- **EJS:** Express JavaScript
- **SOA:** Service-Oriented Architecture
- **WASCE:** (Web Sphere Application Server Community Edition) Web server.

1.6 OVERVIEW

The SRS includes two sections,

Overall Description: This section will describe significant components of the system, interconnections, and external interfaces.

Specific Requirements: This section will describe the functions of actors, their roles in the system, and the constraints faced by the system.

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

2. OVERALL DESCRIPTION

Overall description of auditor management system describes the major components of the system, inter-connections, and external interfaces.

2.1 PRODUCT PERSPECTIVE

- The auditor management system follows a service-based architecture for easy pluggability and interaction with other services.
- The front-end tier has a web application, which had performed with JavaScript.
- The business components then interact with the IBM DB2 database. Pure XML will utilize as it offers a hierarchical storage mechanism that is easier to use.

2.2 SOFTWARE INTERFACE

- **Front End Client:** Web browser, Operating System (any operating system applicable)
- **Web Server:** cPanel or WASCE, Operating System (any operating system applicable)
- **Database Server:** DB2, Operating System (any operating system is applicable)
- **Back End:** DB2
- **Development Tool:** Visual Studio Code, Windows, Web Sphere server

2.3 HARDWARE INTERFACE

- **Client-side:** Internet Explorer 7.0 and above, Google Chrome, Mozilla Firefox, and Opera
- **Processor:** i3 core and above
- **RAM:** 1GB or above for better experience
- **Disk Space:** 2GB

- **Server Side:** Web server application
- **Processor:** i3 or above
- **RAM:** 2GB or above for better experience
- **Disk Space:** 2GB

18CS2009 – 20CS2001 Software Engineering Lab – URK20CS2001

2.4 PRODUCT FUNCTIONALITIES

- **Administrator Facilities:** The administrator facilities include adding new users, editing user-profiles, and generating reports
- **Auditor facilities:** Auditor's facilities include Risk management, providing opinions, and giving finished reports.
- **Reports:** The Auditor and management reports can generate every week. Also, automatic mail client works on the release to send directly to the Board of Directors.
- **User (Auditor Facilities):** The user can view or edit their profiles, change password, log in and log out.

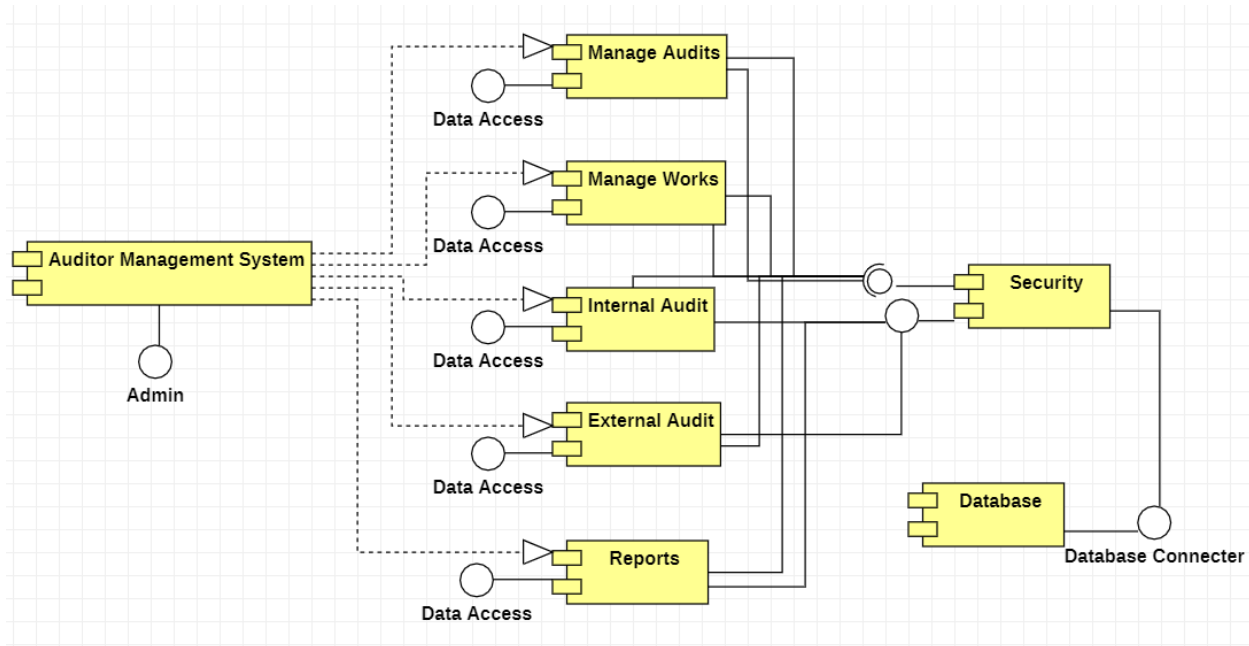
2.5 USER CHARACTERISTICS:

The user must know about operating a computer, browsing the internet and English to login and access the facilities assigned.

2.6 CONSTRAINTS

- GUI is only in English.
- Username and password only used for only the web user authentication
- Only Registered Auditors, managers, company's board of directors are allowed to use the services

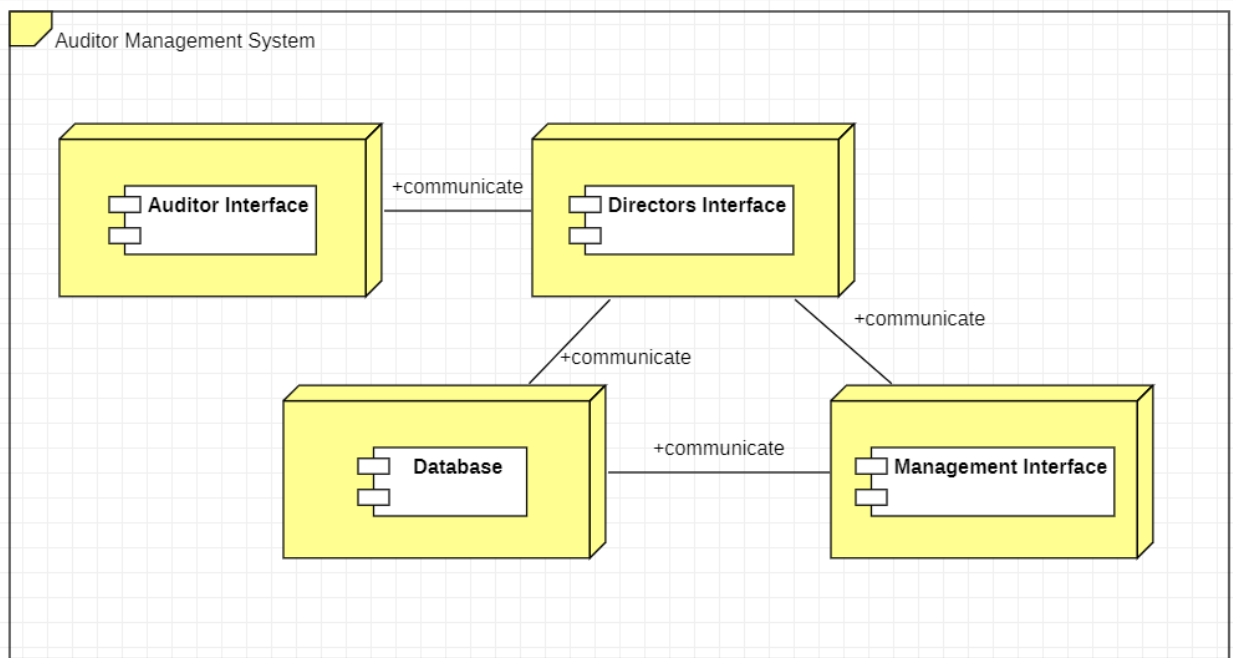
2.7 ARCHITECTURE DIAGRAM



Architecture Diagram is having it's another name of Component diagram which states that the Components which are available in a particular system. Here, in this Auditor management system the component diagram states that the Auditor management system which has connected to the Interface called Admin. And the system has seven components are Manage Audits, Manage Works, Internal Audits, External Audits, Reports, Security, and the Database.

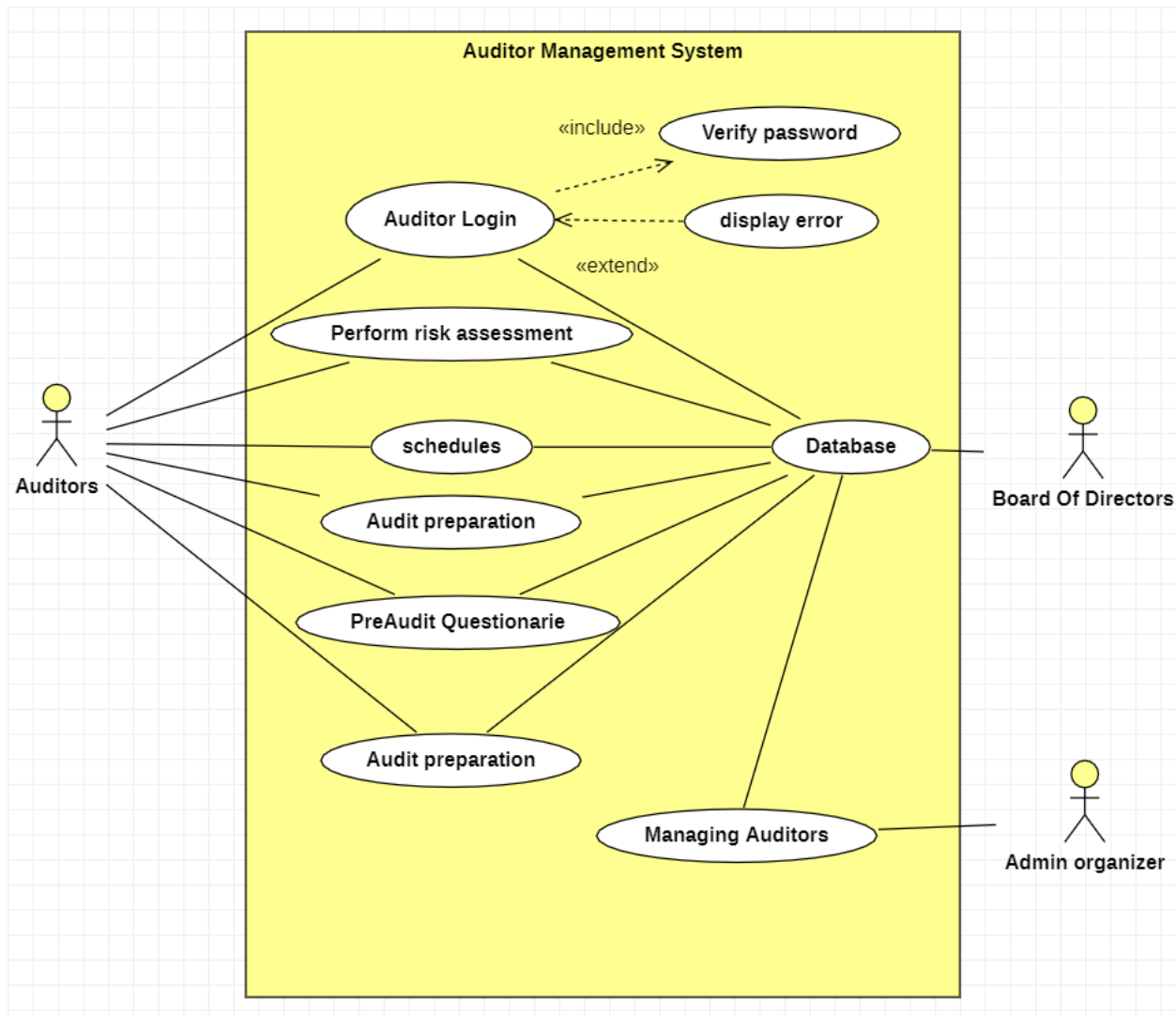
There are two nodes under every component which are available in the main Interface. The main system is connected to the first node of each, and every component and the secondary node is connected to the second node which is the Data Access, and the same nodes are connected to the security of the system with two interfaces. And the last database is connected by using database connector.

2.7.1 DEPLOYMENT DIAGRAM



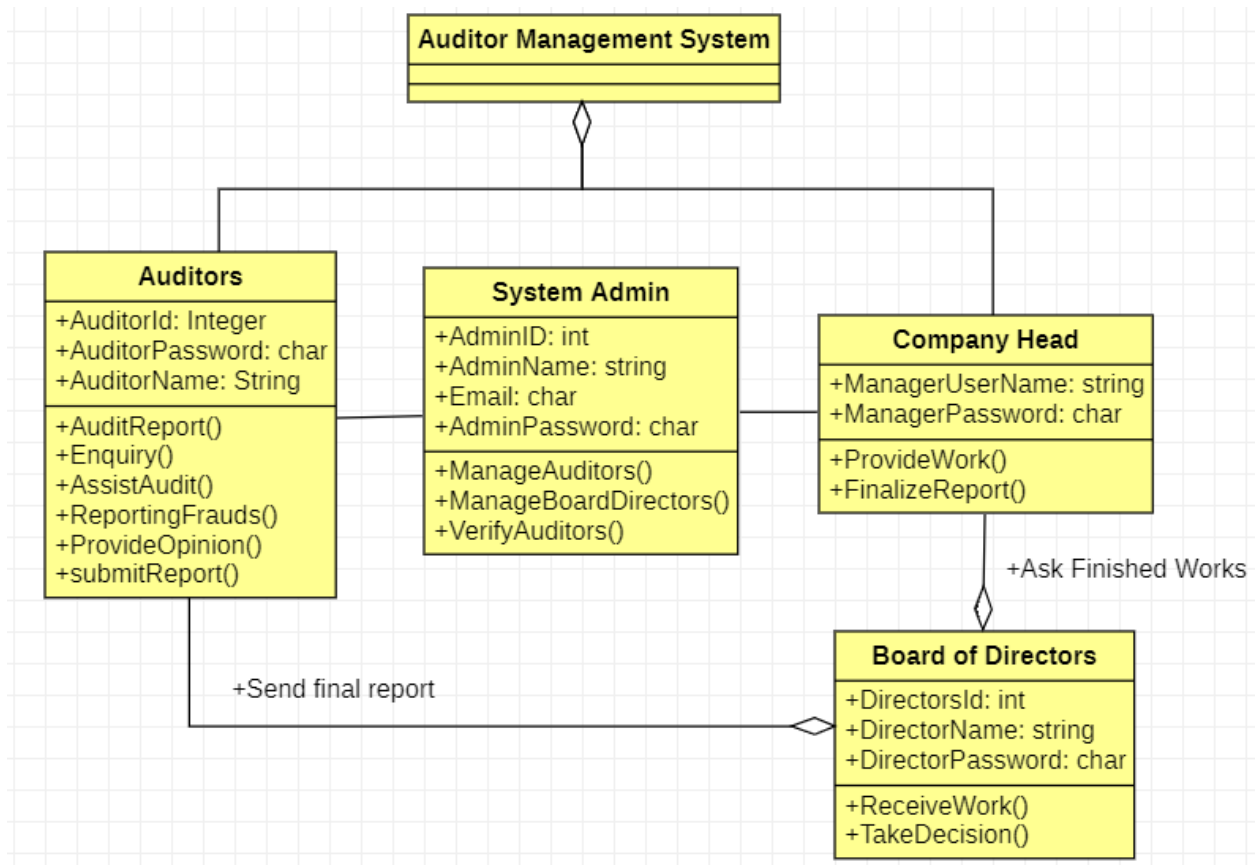
Deployment diagram states that the Interface deployment in the auditor management system. There are four main interfaces in the auditor management system namely, Auditor Interface, Directors Interface, Database, and Management Interface. These Interface are connected with a communication path where the four interfaces of this management system will communicate to the main UI.

2.8 USE CASE MODEL DESCRIPTION



Use case diagram tells us the Main categorial uses of this Auditor Management System. It has three actors where the three actors of Auditors, Board of Directors, and Admin Organizer. Here, Auditors having the use case of Auditor Login, perform risk management, schedules, audit preparation, preaudit questionnaire and audit. All these operations are connected to the Database where the board of directors can easily get the sufficient data's from the database. The Admin panel organizer can easily access these statements at right time to generate the stable Audit reports.

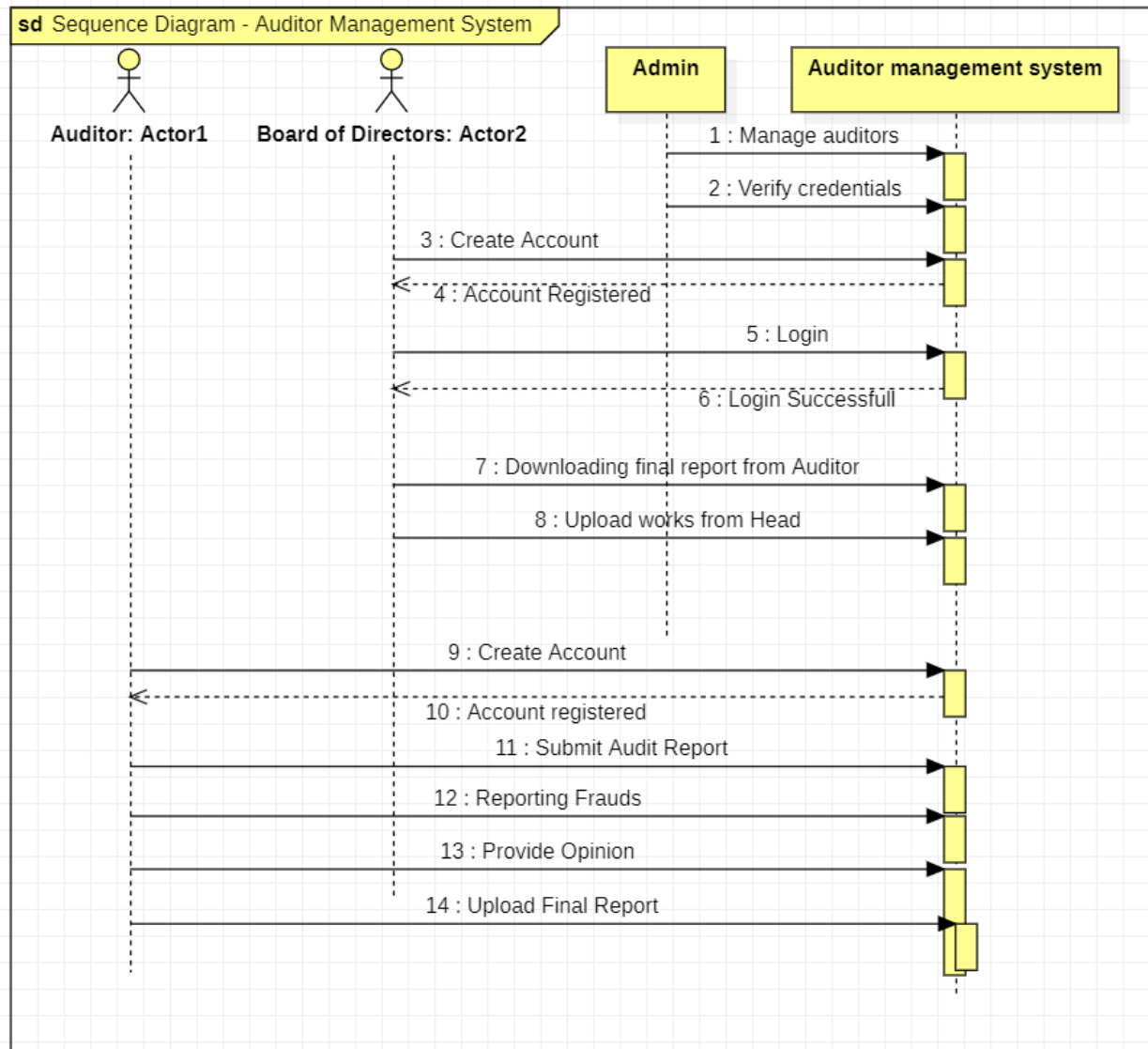
2.9 CLASS DIAGRAM



Class diagrams separate the single system to different classes of Auditors, System Admin, company head and the Board of directors from the auditor management system. Here the auditors have the operations of preparing the Audit report, enquiry, assist audit, reporting frauds, provide opinion, and submit the final report.

System admin Organized the auditors and managing the board of director accounts and verifying the auditors. Here all the classes are connected using aggregation.

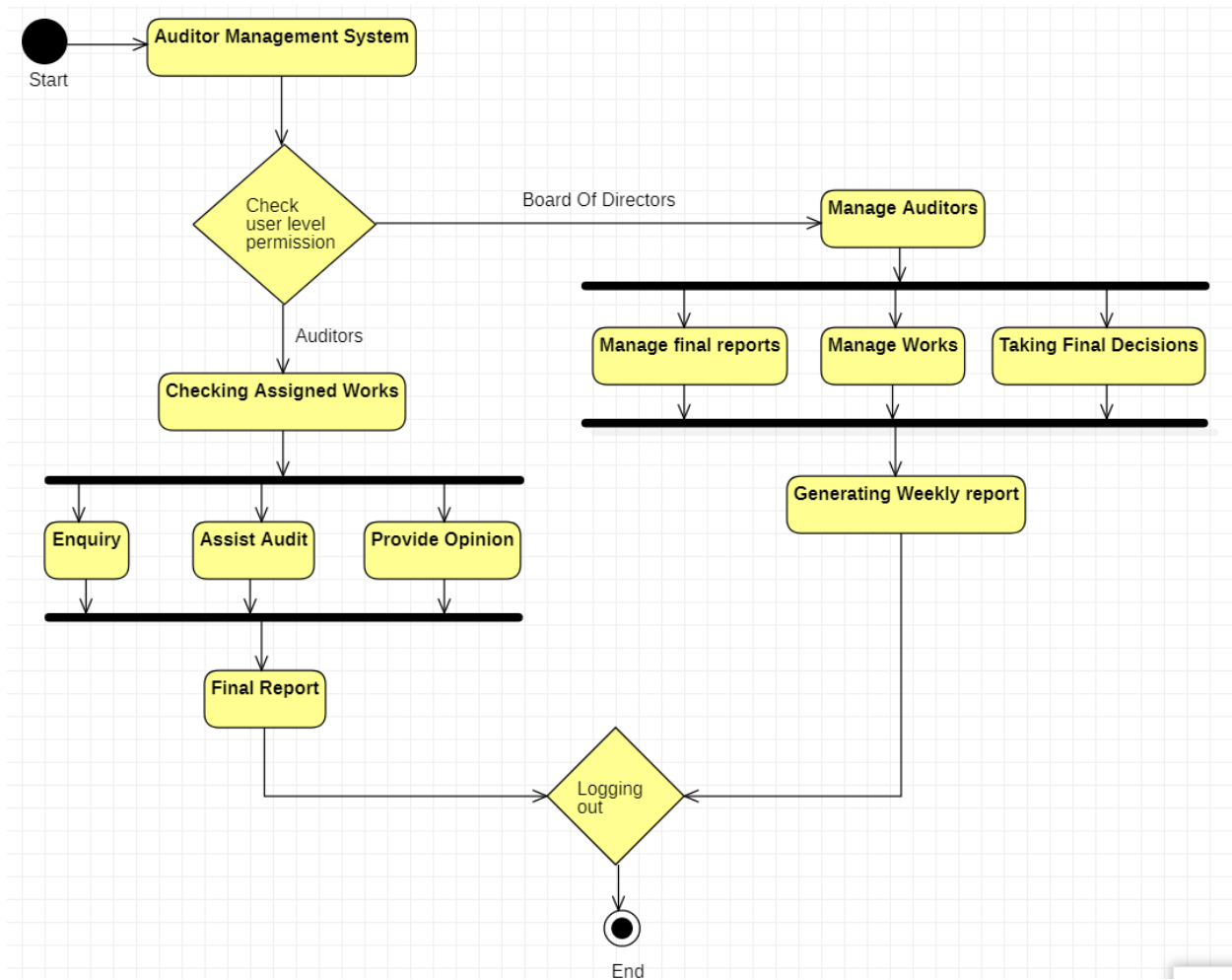
2.9.1 SEQUENCE DIAGRAMS



Sequence Diagram separates the auditor management system into four number of sequences. Here, it has two actors, where first actor is Auditors, and the second actor is board of directors. These two actors are connected into the lifeline of Admin and auditor management system.

First the admin will be verifying the Auditors and the board of directors' accounts. Then the board of directors will take an initiative to create an account and the system will retrieves the message as the account created successful. After the board of directors will download the reports from the auditors. Same things can be done by the auditors. Where they are just upload the final report.

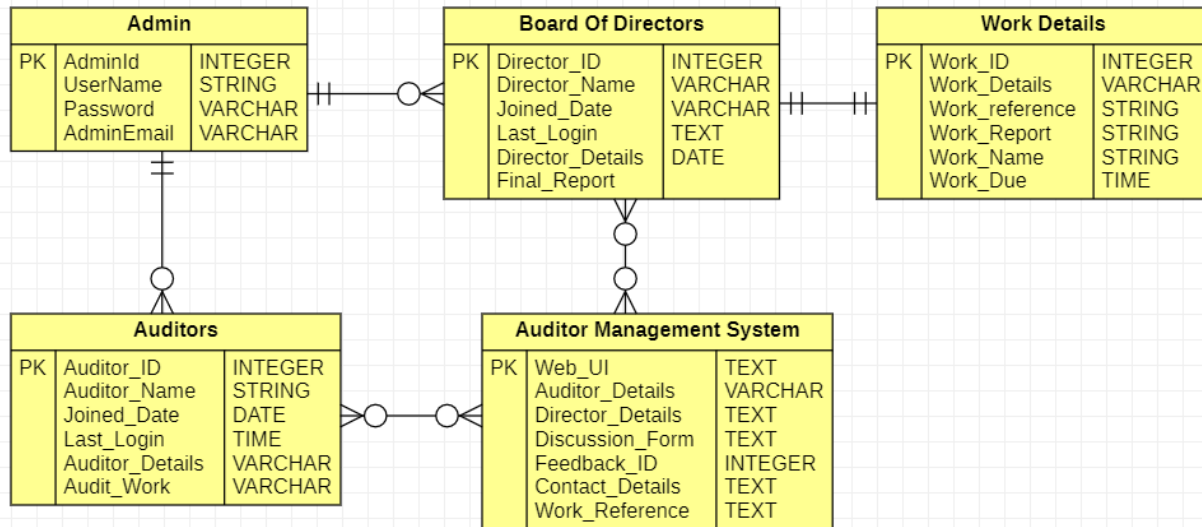
2.10 ACTIVITY DIAGRAM



Activity diagram separates the activities of the system into parts it has start and it check the user level permission. According to the user level permission it will get into the interfaces. In this auditor management system first, it will check the user level permission, and if the user is auditor then it will redirect the user to the auditor interface else it will get the user into the board of director interface. And after both the interface or the works finished in the interface it will proceed to the log out section to end the system.

2.11 DATABASE DESIGN

2.11.1 ER DIAGRAM



ER Diagram is nothing but the Entity Relationship Diagram. Where the Entities are the databases which are in the Auditor Management System, here this system has five databases where the Admin, Auditors, Board of Directors, the Work details are occupies in the Single individual databases.

Admin has the entities of admin id, username, password, and admin email. The Board of directors has the entities of director id, director name, joined date, last login and the director details and the final report. The work details have an entities of work id, work details, work reference, work report, work name, and the work due. The auditors have the entitles of auditor id, auditor name, joined date, last login, auditor details and the audit work.

2.12 ASSUMPTIONS AND DEPENDENCIES

- The controller is created in the system already to add, edit, delete and view the details of the users.
- The users and their roles are predefined.
- The auditors or users of the system should have basic knowledge of English and how to operate a computer and a web browser.

3. SPECIFIC REQUIREMENTS

This auditor management system does not have any specific requirements since the data from the system is fully handled by the cloud

3.1 USE CASE REPORTS

- **Login:** The Auditors, Board of directors and managers can login with username and password.
- **View/Edit profile:** Every individual can be able to edit his/her profile.
- **Assign work:**

3.2 SUPPLEMENTARY REQUIREMENTS

- Secure access of confidential data
- 24/7 availability - The system can be automated so that it works around the clock and during power failures
- Better component design to get better performance at peak time.
- Flexible service-based architecture for future extension

4. TECHNOLOGIES USED

- **HTML:** Hypertext Markup Language
- **CSS:** Cascading Style Sheets
- **JS:** JavaScript
- **JQ:** jQuery
- **XML:** Extensible Markup Language
- **UML:** Unified Modelling Language
- **EJS:** Express JavaScript
- **SOA:** Service-Oriented Architecture
- **WASCE:** (Web Sphere Application Server Community Edition) Web server.

4.1 LANGUAGES USED

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JavaScript: Scripting Language

4.2 IDE/TOOLS USED

VS Code: Visual Studio Code

4.3 API/Framework DOCUMENTATION

Bootstrap: <https://getbootstrap.com/>

IBM DB2: <https://cloud.ibm.com/apidocs/cloudant>

Apache: <http://httpd.apache.org/docs/>

4.4 DEVELOPER'S DOCUMENTATION

HTML: <https://developer.mozilla.org/en-US/docs/Web/HTML>

CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>

JS: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

REST API: <https://developers.google.com/docs/api/reference/rest>

5. LIMITATION OF THE PROJECT

An auditor management system is a heavily used project for big companies so, it wants to be stable and efficient in calculating the risks.

The main limitation would be the time needed to develop this website is around five months. since we have to implement such highly addressable algorithms in Artificial intelligence and some algorithms based on Blockchain technologies. Therefore, the time taken is the only limitation for this project.

RESULT:

A software requirement specification for modelling a prototype of **AUDITOR MANAGEMENT SYSTEM** was carried out successfully.