

QR Code Based Attendance System

LAB REPORT

Submitted by

SWETHA[RA2111026010259]

YUKTHAMUKHI[RA2111026010262]

DHANUSH[RA2111026010248]

RAJANESH[RA2111026010266]

Under the Guidance of

Dr.N.Arivazhagan

Assistant Professor, CINTEL

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

with specialization in Artificial Intelligence and Machine Learning



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY SRM

INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

MAY 2023



COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY

S.R.M. NAGAR, KATTANKULATHUE – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

RegisterNo:**RA2111026010259,RA2111026010262,RA2111026010248,**

RA2111026010266

Certified to be the bonafide work done by **Swetha Suresh, Yukthamukhi Rangineni, Dhanush, Rajanesh** of II Year/IV Sem B.Tech Degree Course in the **Practical Software Engineering and Project Management 18CSC206J** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**,

Kattankulathur during the academic year 2022 – 2023.

FACULTY-IN-CHARGE

Dr.N.Arivazhagan

Assistant Professor
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai

HEAD OF THE DEPARTMENT

Dr. R Annie Uthra

Professor and Head,
Department of Computational Intelligence
SRM Institute of Science and Technology
Kattankulathur Campus, Chennai

TABLE OF CONTENTS

SERIAL NO	TITLE	PAGE NO
1	ABSTRACT	4
2	PROBLEM STATEMENT	7
3	STAKEHOLDERS & PROCESS MODELS	11
4	IDENTIFYING REQUIREMENTS	13
5	PROJECT PLAN	16
6	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	33
7	UML DIAGRAMS	42
8	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	52
9	TEST CASES & REPORTING	64
10	ARCHITECTURE/DESIGN/FRAMEWORK/IMPLEMENTATION (CODE & UI)	70
11	CONCLUSION	106
12	REFERENCES	106

1. ABSTRACT

The project is aimed at developing a QR code-based attendance system that can automate the process of taking attendance in schools, colleges, and organizations. The system is designed to generate a unique QR code for each student or employee, which can be scanned using a mobile phone or any QR code reader. The data is recorded and stored in a database, which can be accessed by authorized personnel for monitoring and analysis purposes.

The proposed QR code-based attendance system offers several advantages over the traditional paper-based attendance system. It eliminates the need for manual attendance marking, thereby reducing errors and saving time. The system provides real-time attendance tracking, automatic alerts for absentees, and easy integration with existing student/employee management systems. Moreover, the system is secure, reliable, and easy to use.

In conclusion, the QR code-based attendance system project can help educational institutions and organizations improve their attendance tracking process. By automating the attendance process, the system can save time, reduce errors, and improve data accuracy. With the increasing use of mobile phones and QR code readers, the proposed system can be easily implemented and used by schools, colleges, and organizations of all sizes.

In conclusion, the proposed QR code-based attendance system project can revolutionize the way attendance is tracked in educational institutions and organizations. By automating the attendance tracking process, the system can save time, reduce errors, and improve data accuracy. The project's success will depend on its ability to address the unique needs of different institutions and organizations, making it a highly customizable and adaptable solution.

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1	Project Management	17
2	Project Leader	18
3	Project Closure	23
4	Schedule Management	24
5	Timeline Gang Chart	25
6	Schedule Control	27
7	Work Breakdown Structure	35
8	Swot	39
9	Architecture Diagram	41
10	Usecase Diagram	41
11	Class Diagram	42
12	ER Diagram	43
13	DFD Level-0	47
14	DED Level-1	48
15	Sequence Diagram	49
16	Collaboration Diagram	51
17	Selenium	82

18	Security Testing	82
19	Admin's Screen UI	104
20	Student's Screen UI	106

2] PROBLEM STATEMENT

Aim

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the attendance system using QR system

Project Title:

QR code based attendance system

Project Description:

A QR code-based attendance system is a method of tracking attendance using Quick Response (QR) codes. In this system, students or employees are required to scan a unique QR code using a smartphone or other device with a QR code reader, in order to mark their attendance.

The QR code is generated and displayed on a computer or projector in the classroom or office, and can be scanned using the camera on a smartphone or tablet. Once the QR code is scanned, the attendance is recorded and the system can confirm the attendance in real-time.

This system is designed to be quick, efficient and easy to use, as well as reducing the need for manual attendance tracking. The use of a QR code also helps to prevent fraud and ensure the accuracy of attendance records.

This system can be used for both in-person and remote attendance, the QR code can be generated on a computer or mobile and sent to the student or employee and they can scan the code from their device from anywhere.

QR code-based attendance systems can also integrate with other software, such as Learning Management Systems (LMS) or Human Resource Management Systems (HRMS), to automatically update attendance records and provide reports on attendance.

Business Case:

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

A QR code-based attendance system aims to solve several problems associated with traditional attendance tracking methods.

- One problem it addresses is the time and effort required for manual attendance tracking. In traditional systems, teachers or managers must manually take attendance by calling out names or collecting paper sign-in sheets, which can be time-consuming and error-prone. With a QR code-based system, attendance can be tracked quickly and easily by simply scanning a code.
- Another problem it addresses is the potential for attendance fraud or errors. In traditional systems, students or employees can cheat or falsify attendance records, which can lead to inaccurate attendance data. With a QR code-based system, each person's attendance is recorded using a unique code, making it more difficult to cheat the system.
- Additionally, this system can help to create an opportunity to improve attendance tracking for remote or hybrid learning systems, during the pandemic or any other situations where most of the student or employee work remotely. With a QR code-based system, students or employees can mark their attendance from anywhere, as long as they have access to a device with a QR code reader.

THE HISTORY

In bullet points, describe the current situation.

- The use of QR code-based attendance systems has become increasingly popular in recent years, especially in the wake of the COVID-19 pandemic. With the need for social distancing and remote learning, many educational institutions and organizations have turned to QR code-based systems as a way to track attendance while minimizing close contact.
- In educational institutions, QR code-based attendance systems are being used in both in-person and remote learning settings. In the classroom, teachers can display a QR code on a computer or projector, and students can use their smartphones to scan the code and mark their attendance. In remote learning settings, teachers can send QR codes to students via email or messaging apps, and students can scan the codes from their own devices to mark their attendance.
- In organizations, QR code-based attendance systems are being used to track employee attendance, especially for remote working or for employees who are rotating and not always present in the office.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- Lack of access to technology: For the system to be successful, students or employees need to have access to a device with a QR code reader, such as a smartphone or tablet. If a significant number of students or employees do not have access to this technology, the system may not be effective.
- Poor internet connectivity: In a remote learning setting, if the internet connectivity is poor, students may not be able to scan the QR code, or the system may not be able to confirm attendance in real-time.
- Difficulty in scanning the code: If the QR code is not clearly visible or is too small, students or employees may have difficulty scanning it, which can lead to inaccurate attendance records.

APPROACH

List what is needed to complete the project.

- QR code generation software: This software is used to generate unique QR codes for each student or employee.
- QR code reader: This can be a smartphone app or a separate device that can read and scan QR codes.
- Database: A database is needed to store attendance records and track attendance for each student or employee.
- Server: A server is needed to host the database and the attendance system software, and to manage the attendance data.
- Network infrastructure: A stable and secure network infrastructure is needed to connect the devices that are used to scan the QR codes to the server and database.
- User interface: A user interface (UI) is needed to display the QR code, to allow users to mark their attendance and to provide an easy way to access the attendance records.
- Technical support: Technical support is needed to troubleshoot any issues that arise with the system and to provide training to users on how to use the system.
- Integration: If needed, the system should be integrated with other systems such as Learning Management Systems (LMS) or Human Resource Management Systems (HRMS) to take the most advantage of the data and to automate the process.
- Security: The system should be designed with security in mind to protect the privacy and confidentiality of student or employee data.
- Testing and Quality assurance: Adequate testing and quality assurance should be done to ensure that the system is working correctly and that attendance records are accurate.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- Convenience: The system allows for easy and quick attendance tracking, which can save time and increase efficiency for both staff and students.
- Contactless Attendance: It minimizes close contact and reduces the spread of germs, which can help to keep employees and students safe and healthy, especially during the

pandemic or flu seasons.

- Real-time tracking: The system allows for real-time tracking of attendance, which can help to quickly identify and address any attendance issues.
- Remote Attendance: It enables remote attendance tracking which can be very useful for remote working employees and also for organizations with employees working in different locations.
- Accurate attendance records: The system can provide accurate attendance records, which can be used for compliance, payroll and other administrative purposes.
- Integration: It can be integrated with other systems such as Learning Management Systems (LMS) or Human Resource Management Systems (HRMS) which can automate the process and increase the effectiveness of the system.
- Cost-effective: The system can be a cost-effective solution for attendance tracking, as it eliminates the need for manual attendance tracking and reduces the need for expensive equipment.
- Better data analysis: The system can provide data that can be analyzed to identify patterns, trends, and areas for improvement, which can help to improve overall organizational performance.

Result

Thus, the project team formed, the project is described, the business case was prepared and the problem statement was arrived.

EXPERIMENT-02

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Project Title: Attendance system using QR-code

Agile Methodology

the Agile methodology is known for its flexibility, whereas Waterfall is a structured software development methodology.

- Waterfall is a Linear Sequential Life Cycle Model, whereas Agile is a continuous iteration of development and testing in the software development process.
- Agile performs testing concurrently with software development, whereas in Waterfall methodology, testing comes after the “Build” phase.
- Agile allows changes in project development requirements, whereas Waterfall has no scope of changing the requirements once the project development starts.
- Comparing the Waterfall methodology vs Agile, which follows an incremental approach, whereas the Waterfall is a sequential design process.

Incorporate information to below table regarding stakeholders of the project

[Make use of below examples]

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
End-users	Employees or students who use the QR code to mark their attendance.	High	High	4
System administrators	Those responsible for	High	Low	5

	managing the QR code attendance system, including setup and maintenance.			
IT department	They are responsible for ensuring the technology infrastructure is in place to support the QR code system.	Medium	Medium	5
Management	They make decisions regarding the implementation and usage of the QR code system.	Medium	Medium	4
External vendors	Companies that provide the QR code technology or related services.	Low	High	3

Result

Thus the Project Methodology was identified and the stakeholders were described.

EXPERIMENT-03

Aim

To identify the system, functional and non-functional requirements for the project.

Project Title: Attendance system using QR-code

System Requirements:

Software requiements

- Android Studio
- Android OS 4.2+ (Mobile)
- MS Excel

Hardware requirement

- Android Smartphone running Android OS version 4.0.3(API level 15)+

Input/output

- Input: QR code, Bar code
- Output: XLS and CSV sheet

Database used

- SQLite Database

Functional Requirements

1. QR code generation and distribution:
 - The system should allow for the generation of unique QR codes for each student or attendee
 - The system should be able to distribute the QR codes to students or attendees in a convenient and secure way, such as through email or a secure online portal
2. Attendance tracking:

- The system should be able to track attendance in real-time based on the scans of the QR codes
 - The system should be able to record the time and date of the attendance scan
 - The system should be able to store attendance data securely and be able to retrieve it easily
3. Reporting and analysis:
 - The system should be able to generate reports that show attendance data, such as who attended and when they attended
 - The system should be able to generate reports that show attendance trends over time, such as attendance rates by day, week, or month
 - The system should be able to provide insights and analytics to help users make informed decisions about attendance management
 4. Administration:
 - The system should be able to allow for the creation and management of classes, courses, or events
 - The system should be able to allow for the management of student or attendee information, such as adding or removing students/attendees, updating contact information, or managing attendance exceptions
 5. Integration and compatibility:
 - The system should be able to integrate with other software or systems, such as learning management systems, student information systems, or human resource management systems
 - The system should be compatible with a variety of devices and platforms, such as iOS and Android mobile devices, web browsers, or desktop computers

Non-Functional Requirements

1. Performance:
 - The system should be able to handle a high volume of attendance scans without significant delays or performance issues
 - The system should be able to generate attendance reports quickly and accurately
 - The system should be able to operate reliably and consistently over time
2. Security:
 - The system should provide secure access to attendance data, ensuring that only authorized users can view or modify attendance records
 - The system should use encryption and other security measures to protect data in transit and at rest
 - The system should provide data backup and recovery capabilities to ensure that attendance data is not lost in the event of a system failure or other disaster
3. Usability:
 - The system should be easy to use and navigate for both instructors and students/attendees
 - The system should have a user-friendly interface and provide clear instructions for QR code scanning and attendance tracking
 - The system should minimize the need for user training or technical support
4. Compatibility:
 - The system should be compatible with a variety of devices and platforms, such as iOS and Android mobile devices, web browsers, or desktop computers

- The system should be able to integrate with other software or systems, such as learning management systems, student information systems, or human resource management systems
5. Scalability:
- The system should be able to scale up or down as needed to accommodate changes in the number of students/attendees or classes/courses/events
 - The system should be able to handle increases in attendance data volume without significant performance or usability issues

Result

Thus the requirements were identified and accordingly described.

EXPERIMENT-04

Aim

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

Team Members:

Sl No	Register No	Name	Role
1	RA2111026010266	Rajanesh	Rep/Member
2	RA2111026010262	Yukthimukhi	Member
3	RA2111026010259	Swetha	Member
4	RA2111026010248	Dhanush	Member

Project Title: QR-code Based Attendance System

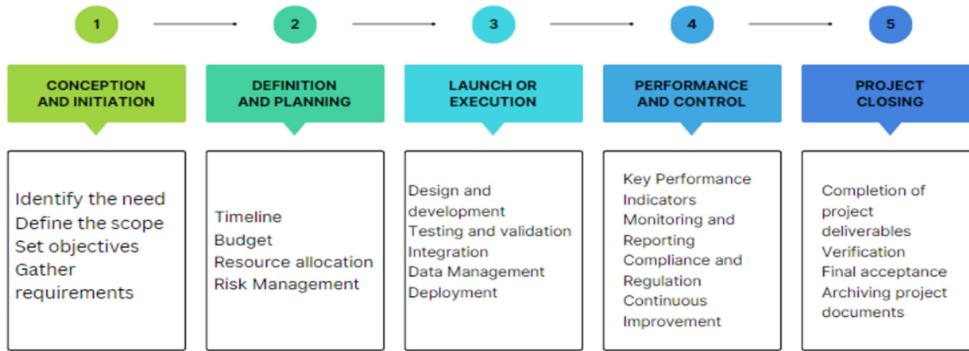
Project Plan

Integration Management

Governance Framework :

The governance framework acts as an essential supporting structure, a framework of rules and practices by which the board ensures accountability, fairness and transparency in how the company runs and communicates with its stakeholders. The governance framework in project management typically includes the following components:

Project Management



Project management (figure 1)

Project governance structure:

This includes the roles and responsibilities of project stakeholders, such as project sponsors, steering committees, project managers, and project teams. It outlines how decisions are made, who has the authority to make them, and how they are communicated.

Project management processes:

This includes the processes and procedures for planning, executing, monitoring, and controlling projects. It outlines how project objectives are defined, how resources are allocated, and how project progress is monitored and reported.

Project performance metrics:

This includes the measures used to evaluate project performance, such as cost, schedule, quality, and risk. It outlines how project performance is tracked and how corrective actions are taken when needed.

Project management tools and technology:

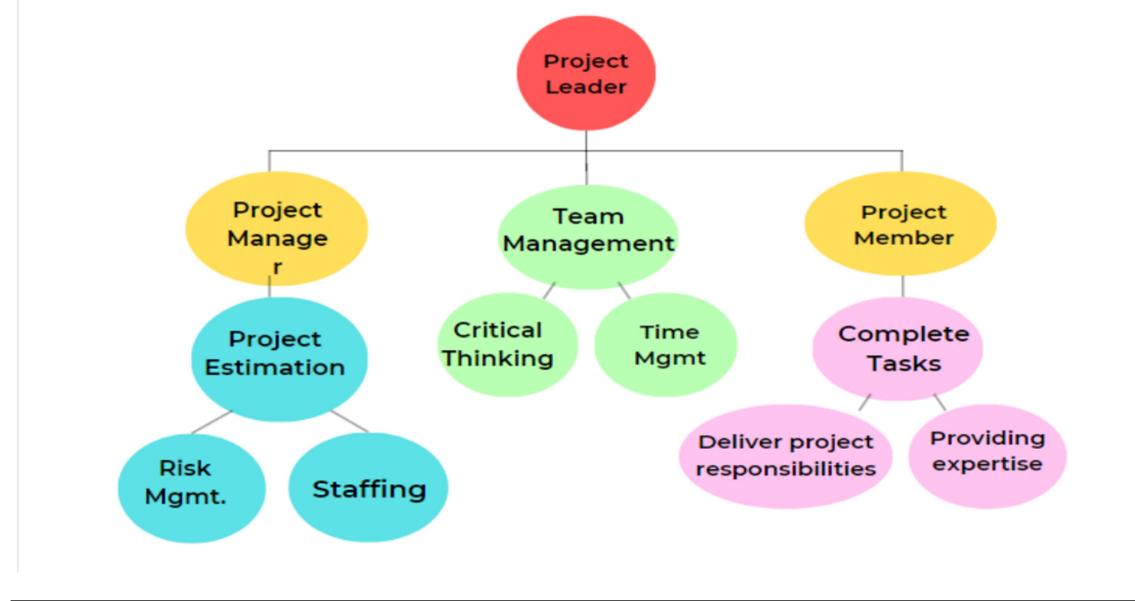
This includes the tools and technology used to manage projects, such as project management software, communication tools, and collaboration platforms. It outlines how these tools are used to facilitate project management processes.

Project management standards and best practices:

This includes the standards and best practices for project management, such as PMI's Project Management Body of Knowledge (PMBOK) and the PRINCE2 methodology. It outlines how these standards and best practices are applied to projects within the organization.

Project Team structure, their roles and responsibilities

Project Leader (figure 2)



Project Leader

- Develop team schedules and assist in the successful onboarding and training of team members

- Create and communicate a clear list of expectations and goals for team members to follow
- Offer emotional support to project team members and make people feel valued

Project Manager

- Planning everything from execution to delivery
- Directing the team to achieve a common goal
- Delegating work effectively
- Managing the resource of time
- Managing documentation and reports

Project Member

- Complete the project objectives
- Members of a project team can have various roles.
- Delegating work effectively
- Work together to complete a project or goal

Cost Management

Estimate Effort:

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981. COCOMO is one of the most widely used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

In COCOMO, projects are categorized into three types:

1. Organic
2. Semi Detached
3. Embedded

Organic:

A development project can be treated as organic if it deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

Estimated Number of Lines of Code (SLOC) : **2500 SLOC**

$$\text{Effort} = a(k\text{LOC})b$$

$$\text{Development Time} = c(\text{Effort})d$$

$$\text{Effort} = 2.4(2.5)1.05$$

$$= 2.4(2.617)$$

$$= 6.281 \text{ persons month}$$

Assign Team :

1. RAJANESH - PROJECT LEADER
2. DHANUSH - PROJECT MANAGER
3. SWETHA - PROJECT MEMBER
4. YUKTHAMUKHI –PROJECT MEMBER

Budget Control :

$$\text{Dev. Time} = 2.5(6.28)0.38$$

$$= 2.5(2.01)$$

$$= 5.02 \text{ months}$$

Controlling the budget in a vehicle maintenance system requires careful planning, monitoring, and management. Here are some key strategies to help you keep your project within budget:

Develop a detailed budget plan: Before the project begins, develop a detailed budget plan that outlines all of the costs associated with the project. This plan should include all project costs, such as labor, hardware, software, training, travel, and any other expenses. The plan should be based on realistic estimates of the costs of the project and should be periodically reviewed and updated throughout the project.

Monitor spending: Monitor spending closely throughout the project to ensure that costs stay within budget. Track spending on a regular basis and compare actual costs to the budgeted costs to identify any areas where spending is

exceeding the budget. This will enable the project manager to take corrective action to bring spending back within budget.

Use cost-saving measures: Implement cost-saving measures where possible to reduce project costs. This may involve negotiating with vendors for better pricing, using open-source software instead of commercial software, using cloud-based resources instead of on-premises resources, and minimizing travel and other expenses.

Manage project scope: Ensure that the project stays within scope and does not expand beyond the original requirements. Scope creep can cause costs to increase, so it is important to manage the scope carefully and to avoid adding unnecessary features or requirements to the project.

Optimize resource utilization: Optimize the utilization of resources, including labor, equipment, and materials, to ensure that they are being used efficiently and effectively. This may involve adjusting schedules or shifting resources to different areas of the project to avoid bottlenecks and delays.

Use project management software: Use project management software to track spending, manage resources, and monitor progress. This will enable the project manager to identify potential budget issues before they become significant problems and to take corrective action to keep costs under control.

Regularly review and adjust the budget: Regularly review and adjust the budget throughout the project to ensure that it remains accurate and realistic. This will enable the project manager to make adjustments to the budget as needed to accommodate changes in project scope, resource requirements, or other factors.

In summary, controlling the budget in a QR based Attendance system requires a proactive and vigilant approach to planning, monitoring, and management. By developing a detailed budget plan, monitoring spending, using cost-saving measures, managing project scope, optimizing resource utilization, using project management software, and regularly reviewing and adjusting the budget, it is possible to keep costs under control and ensure the success of the project.

Change Management:

Change management in software development projects is the process of identifying, preparing and supporting the implementation of software changes.

It is applied throughout the whole software development process. New requirements and the need for changes may appear suddenly and can shift several times. If you don't manage them well, your project may be at risk of failing.

Each software development project is defined by three dimensions: scope, time and budget. Most project changes affect some or all of these values. Also, every change poses a certain risk to the implementation of the software project you are developing. You can avoid or minimize risk but sometimes you just have to accept it. The key is to be aware of the full implications of each change for the software project you are working on. To deal with change management in the best possible way, stick to a consistent change management plan.

Some modules/ prerequisites cannot be changed in our software are:

- Users Login
- Interface and functionality
- Customizable notifications and reminders
- User Details

Project Closure(figure 3)



Project Closure:

Project closure is a critical step in the life cycle of any project, including the implementation of a QR-based attendance system. The following are some key steps that can be taken to ensure a successful project closure for a QR-based attendance system:

1. Conduct a final review: Conduct a final review of the project to assess its overall success and identify any remaining issues or tasks that need to be addressed before closure.
2. Close out any outstanding tasks: Close out any remaining tasks or issues identified in the final review. This may include resolving any outstanding technical issues or completing any outstanding training or support tasks.
3. Obtain feedback: Obtain feedback from stakeholders, including employees and managers, about the effectiveness of the QR-based attendance system. This feedback can be used to improve the system or inform future projects.
4. Document lessons learned: Document any lessons learned during the project, including what worked well and what could be improved in future projects.
5. Archive project documentation: Archive all project documentation, including plans, schedules, reports, and other project artifacts. This will ensure that the information is available for future reference or for auditing purposes.
6. Celebrate success: Celebrate the successful implementation of the QR-based attendance system with the project team and stakeholders. This can include recognizing the contributions of team members, highlighting key achievements, and sharing any positive feedback received from employees or other stakeholders.

By following these steps, organizations can ensure a smooth and successful closure of a QR-based attendance system project, while also capturing key learnings that can be applied to future projects.

Schedule Management(**figure 4**)



Schedule Management

1. Defining Milestones

A milestone is a specific point within a project's life cycle used to measure the progress toward the ultimate goal. Milestones in project management are used as signal posts for a project's start or end date, external reviews or input, budget checks, submission of a major deliverable, etc. A milestone is a reference point that marks a significant event or a branching decision point within a project.

For our projects we have defined various tasks/ modules and milestones to be achieved in between. Some of the tasks have already been completed while some are ongoing. These all are represented on a timeline as follows using Ganttpro software:



Timeline gang chart(figure 5)

2. Schedule Control

Schedule control is used in project management to monitor the activities and tasks to ensure you're proceeding as planned. Of course, it's more than just

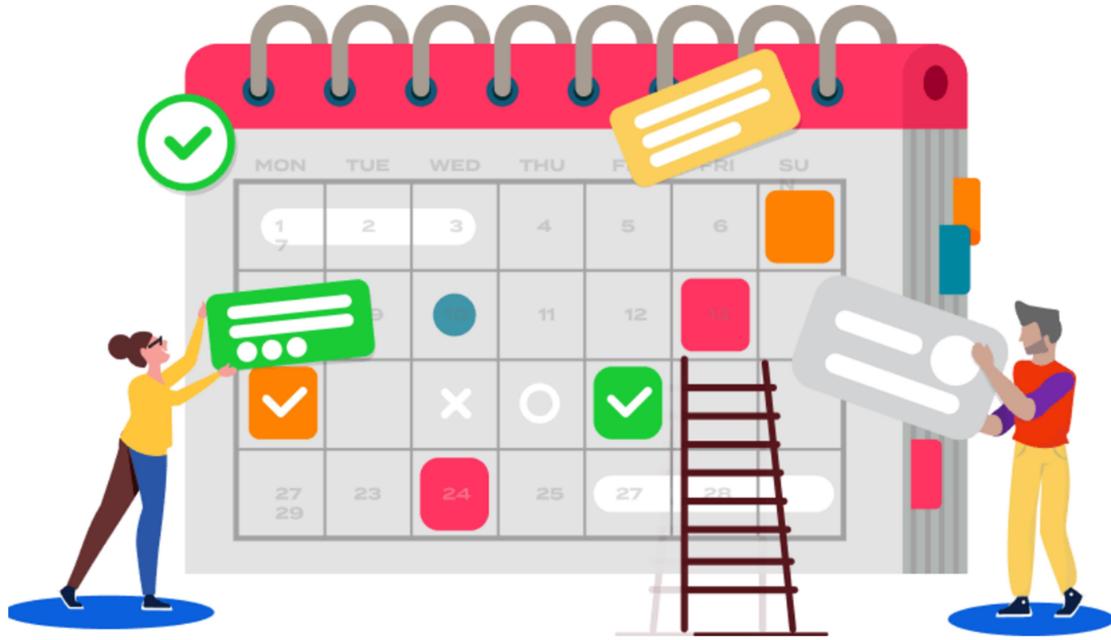
monitoring status. Project schedule control also means updating your project processes and managing change.

The most common way to do this is by creating a baseline. A baseline represents how you expect the project to proceed. When you have a schedule baseline, you can look at your actual progress and compare it to the planned progress, which lets you know if you're ahead or behind schedule. This whole process is simplified with the help of project scheduling software.

Project schedule control is essential because it is used to monitor and control the progress of the project. A project that misses its deadline is not a success. While there are many metrics to measure the success of a project, keeping to the schedule is among the most fundamental.

Another important aspect of schedule control is that it manages the expectations of your project stakeholders. It keeps them informed of changes and how they will impact the project. Furthermore, it tells the stakeholders how you're going to respond to issues to get the project back on track. This process is key to keeping stakeholders in the loop and should be done throughout the life cycle of the project.

Project planning, scheduling and control work hand-in-glove to reduce the impact on your budget. The benefit is financial stability, which projects need to deliver at a cost that has been approved by stakeholders. It also creates documentation that can be helpful when researching historical data for future projects similar to the one you're managing currently.



Schedule Control(figure 6)

Resource Management

People and skills:

Developing QR based Attendance System software can be a complex process that requires a range of skills and expertise. Here are some of the people and skills that may be required to create a vehicle maintenance software:

1.Knowledge of vehicle maintenance processes:

In order to create effective vehicle maintenance software, you will need to have a good understanding of the maintenance processes used in the automotive industry. This includes knowledge of common maintenance tasks, recommended maintenance schedules, and best practices for vehicle care.

2.Programming skills:

You will need strong programming skills in order to create software that is functional, user-friendly, and reliable. The programming languages commonly used in software development include Java, Python, C++, and C#.

3.Knowledge of database design:

Vehicle maintenance software will need to store data such as vehicle information, service history, and maintenance schedules. You will need to be familiar with database design principles and tools such as SQL.

4.Collaboration and communication:

You will likely be working as part of a team, so good collaboration and communication skills are essential. This includes the ability to work well with others, communicate effectively, and be open to feedback and suggestions.

5.Project Managers:

They oversee the development of the software, ensuring that it is delivered on time and within budget.

6.Technical Writers:

They create documentation that explains how to use the software and provide users with a guide to troubleshooting common issues.

7.Quality Assurance (QA) Testers:

They test the software to identify and resolve any bugs, issues, or errors.

Budget Required:

The cost of developing QR based Attendance system software can vary depending on various factors such as the complexity of the software, features and functionalities required, development platform, and the hourly rate of the development team. It is difficult to provide a specific cost without more information about the project requirements.

However, to give you a general idea, the development of QR code based attendance system in India may cost anywhere from INR 2,00,000 to INR 10,00,000 or more. This is just an estimate, and the actual cost may be higher or lower depending on the factors mentioned above. It's recommended to consult with a software development company to get a more accurate estimate based on your specific requirements.

Facilities Required:

Making vehicle maintenance software would require several facilities to ensure that the software functions as intended and provides a high-quality user experience. Here are some of the key facilities that may be required:

1. Development Environment:

Developers will need access to a development environment that includes a code editor, debugging tools, and access to necessary libraries and APIs to develop the software.

2. Database Management System:

A database management system is required to store data related to vehicle maintenance, such as maintenance schedules, vehicle details, and service records.

3. Testing Environment:

A testing environment is required to ensure that the software functions correctly and is free of bugs. This may include tools for automated testing and manual testing.

4. Security Measures:

Security measures should be implemented to protect the sensitive data stored in the software, such as user login information and vehicle maintenance records.

5. Technical Support:

Technical support should be available to users to provide assistance with any issues they may encounter while using the software. Overall, making vehicle maintenance software requires a combination of development, design, testing, deployment, and support facilities to ensure that the software is functional, user-friendly, and secure.

Risk Management

SWOT ANALYSIS:

A SWOT analysis is a useful tool for assessing the strengths, weaknesses, opportunities, and threats of a business or product. Here is a SWOT analysis of a QR code-based attendance system:

Strengths:

- Easy to use and convenient for both employees and administrators

- Accurate tracking of attendance data and reduction of errors
- Can be integrated with other software, such as payroll or time tracking systems
- Can save time and reduce paperwork compared to traditional attendance systems

Weaknesses:

- Requires access to a smartphone or other device with a QR code scanner
- May not be suitable for all industries or work environments
- Relies on employees to remember to scan the QR code, which can lead to incomplete data

Opportunities:

- Can be used for contactless attendance tracking, which is particularly important during the COVID-19 pandemic
- Can provide valuable data for HR and management, such as attendance trends and employee punctuality
- Can be used for other purposes, such as tracking equipment or inventory

Threats:

- Competing attendance systems may offer similar or better features
- Security risks associated with QR codes, such as malicious codes or phishing attacks, may cause concerns for some users
- May not be accepted or adopted by all employees or organizations.

Overall, a QR code-based attendance system has several strengths, but also some weaknesses and potential threats. However, the opportunities for improving attendance tracking, collecting valuable data, and enhancing workplace efficiency make it a promising tool for many organizations.

Cost Estimation:

For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take. There are several factors on which this depends:

- Project scope must be established in advanced.
- Software metrics are used as a support from which evaluation is made.
- The project is broken into small PCs which are estimated individually.
- To achieve true cost & schedule estimate, several option arise.
- Delay estimation.
- Used symbol decomposition techniques to generate project cost and schedule estimates.
- Acquire one or more automated estimation tools.

Analysing all these various factors, we have developed an estimation based on the requirements and calculated an average cost and approximate effort required as follows:

Domain	Sub-Task	Effort (in hrs)	Cost (in Rs.)
Development Team	<ul style="list-style-type: none"> • Developement of App • Designers • Data Analysts 	1500	75,000
Hardware and Software Requirements	<ul style="list-style-type: none"> • Servers • Networking Equipments • Software Licenses 	750	2,25,000
Testing and Quality Assurance	<ul style="list-style-type: none"> • Testing • Research and Development 	800	2,00,000
Miscellaneous Expenses	<p>Personal Expenses like:</p> <ul style="list-style-type: none"> • Travel • Training • Treatment 	150	50000
Total		3200	5.5 Lakhs

Result:

Thus, the Project Plan was documented successfully.

EXPERIMENT-05

Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table

Project Title: QR-code Based Attendance System

Work Breakdown Structure

Planning Phase

- 1.1 Define project scope and requirements
- 1.2 Conduct feasibility analysis
- 1.3 Identify project deliverables
- 1.4 Develop project schedule
- 1.5 Determine project budget

Analysis and Design Phase

- 2.1 Analyze user needs
- 2.2 Define system architecture
- 2.3 Design database schema
- 2.4 Create wireframes and mockups
- 2.5 Select technologies and tools

Development Phase

- 3.1 Set up development environment
- 3.2 Develop server-side application
- 3.3 Develop client-side application

- 3.4 Integrate QR-code scanning functionality
- 3.5 Implement database connectivity
- 3.6 Implement user authentication and authorization

Testing Phase

- 4.1 Conduct unit testing
- 4.2 Conduct integration testing
- 4.3 Conduct system testing
- 4.4 Conduct user acceptance testing

Deployment Phase

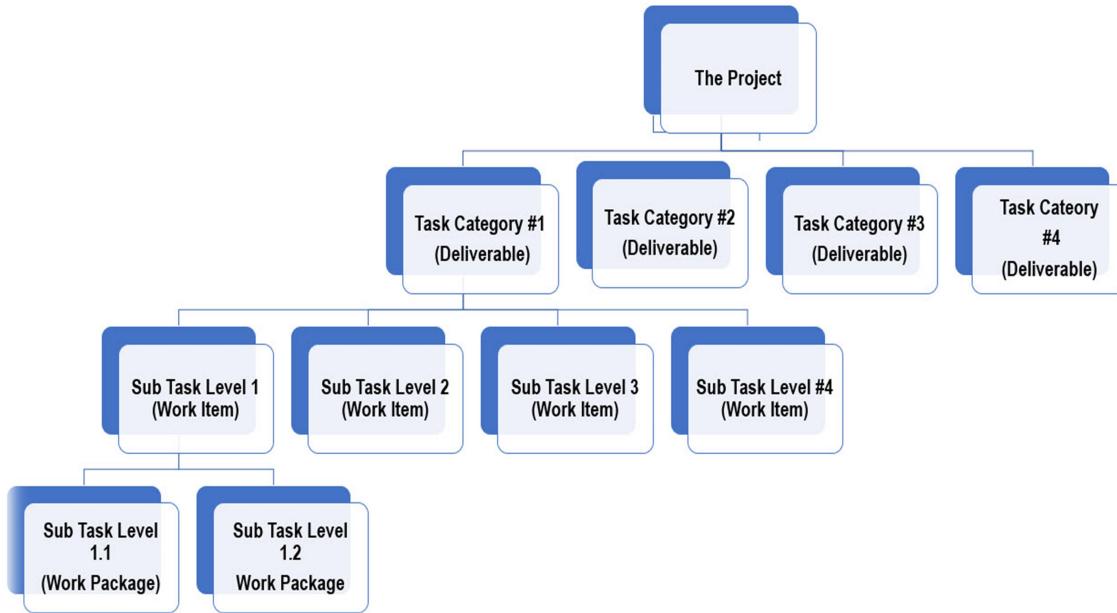
- 5.1 Deploy system to production environment
- 5.2 Provide user training and support
- 5.3 Conduct post-deployment testing
- 5.4 Monitor system performance and user feedback
- 5.5 Address any issues or bugs that arise

Maintenance and Upgrades Phase

- 6.1 Provide ongoing system maintenance
- 6.2 Implement updates and upgrades as needed
- 6.3 Address any security vulnerabilities that arise
- 6.4 Provide technical support to users

This is just a high-level example, but you could break each task down further into more detailed sub-tasks as needed.

Work Breakdown Structure (WBS)



Work breakdown structure(figure 7)

I. Project Planning

- Define project scope and objectives
- Identify stakeholders
- Establish project timeline
- Develop project budget
- Assign project roles and responsibilities

II. Requirements Gathering

- Define functional requirements
- Define non-functional requirements
- Define user requirements
- Define system requirements

III. Design and Development

- Design database schema
- Develop user interface
- Develop backend logic
- Develop reporting system

IV. Testing and Deployment

- Conduct unit testing

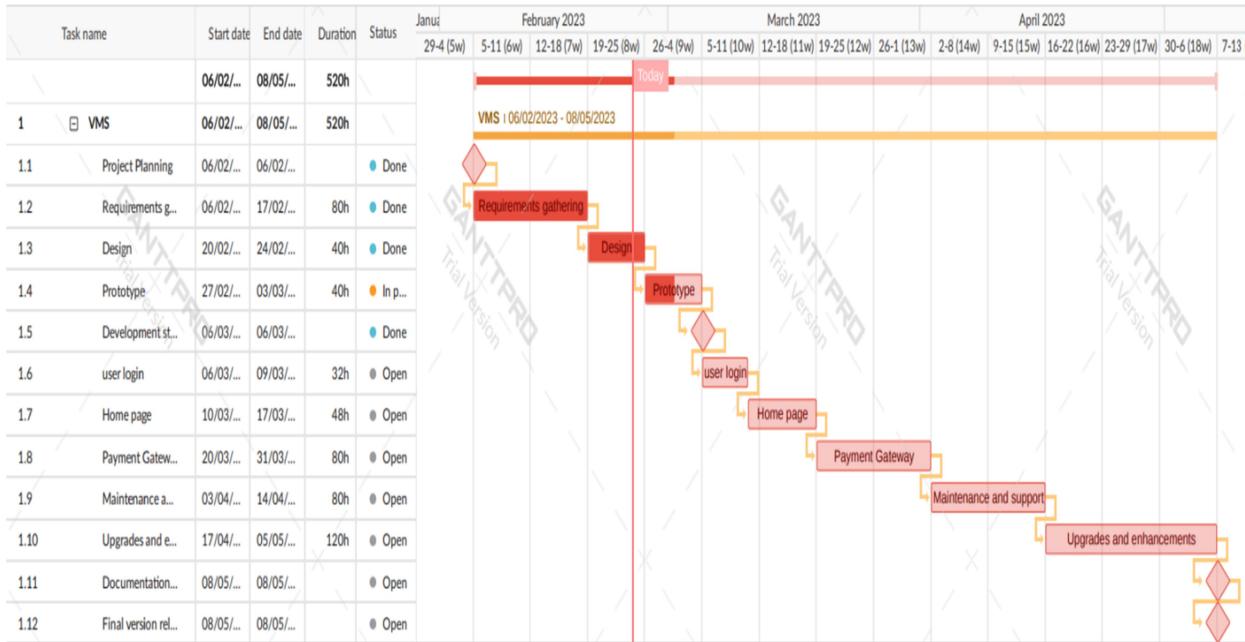
- Conduct integration testing
- Conduct system testing
- Conduct user acceptance testing
- Deploy the system

V. Maintenance and Support

- Provide ongoing technical support
- Perform system maintenance and updates
- Monitor system performance and address issues as they arise

Timeline Chart

Timeline charts are highly versatile visual charts that are used to illustrate a set of events chronologically. They're an excellent tool for conceptualizing event sequences or processes to gain insights into the nuances of a project. That could include summarizing historical events, or any other time frame where you need to measure minutes, hours, dates, or years. A timeline is a chart that depicts how a set of resources are used over time. If you're managing a software project and want to illustrate who is doing what and when, or if you're organizing a conference and need to schedule meeting rooms, a timeline is often a reasonable visualization choice. One popular type of timeline is the Gantt chart.



Risk Management

SWOT ANALYSIS:

A SWOT analysis is a useful tool for assessing the strengths, weaknesses, opportunities, and threats of a business or product. Here is a SWOT analysis of a QR code-based attendance system:

Strengths:

- Easy to use and convenient for both employees and administrators
- Accurate tracking of attendance data and reduction of errors
- Can be integrated with other software, such as payroll or time tracking systems
- Can save time and reduce paperwork compared to traditional attendance systems

Weaknesses:

- Requires access to a smartphone or other device with a QR code scanner

- May not be suitable for all industries or work environments
- Relies on employees to remember to scan the QR code, which can lead to incomplete data

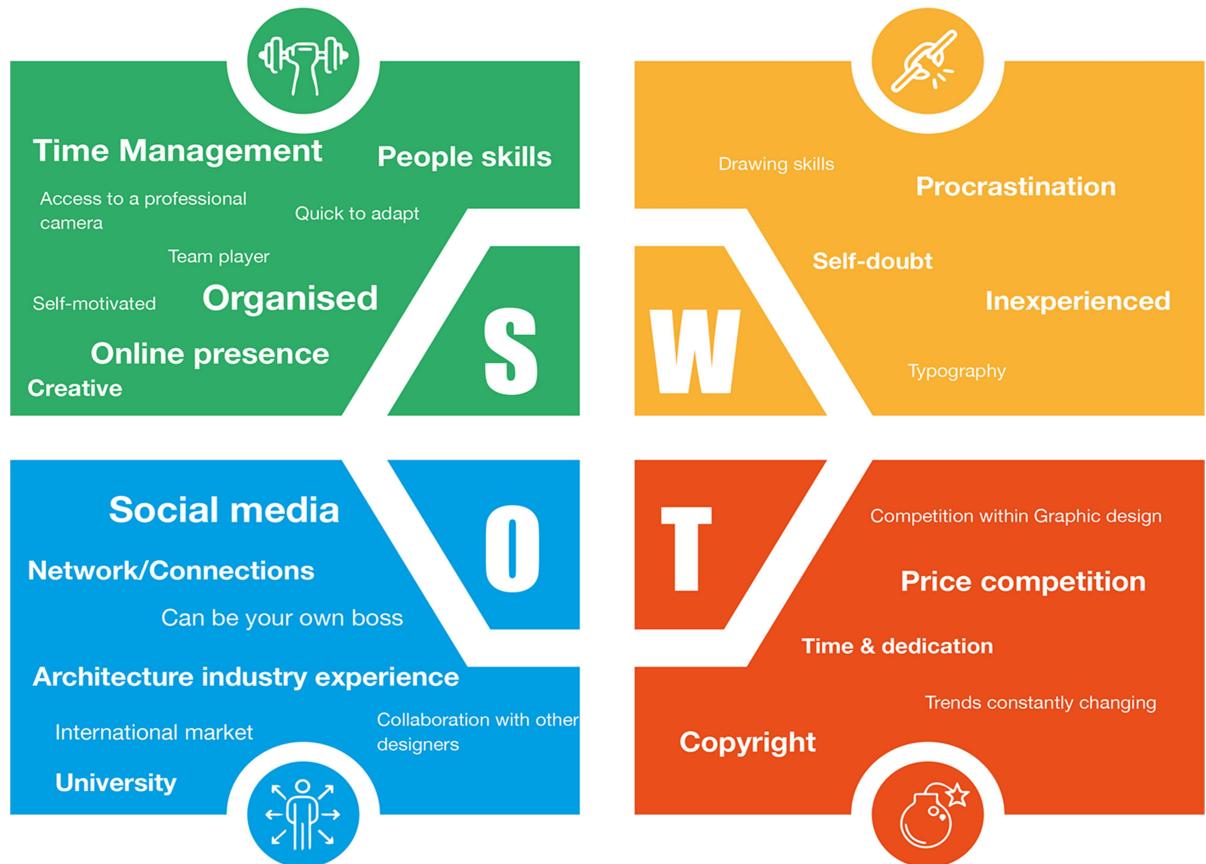
Opportunities:

- Can be used for contactless attendance tracking, which is particularly important during the COVID-19 pandemic
- Can provide valuable data for HR and management, such as attendance trends and employee punctuality
- Can be used for other purposes, such as tracking equipment or inventory

Threats:

- Competing attendance systems may offer similar or better features
- Security risks associated with QR codes, such as malicious codes or phishing attacks, may cause concerns for some users
- May not be accepted or adopted by all employees or organizations.

Overall, a QR code-based attendance system has several strengths, but also some weaknesses and potential threats. However, the opportunities for improving attendance tracking, collecting valuable data, and enhancing workplace efficiency make it a promising tool for many organizations.



Swot(figure 8)

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.

EXPERIMENT-06

Aim

To Design a System Architecture, Use case and Class Diagram

Project Title: QR-code Based Attendance System

System Architecture:

A QR code based attendance system can be implemented using a client-server architecture, where the client is a mobile device with a QR code reader, and the server is a web application that receives and processes the attendance data.

Here's a **high-level** overview of the **system architecture**:

Client-side: A mobile application that generates a unique QR code for each attendee and scans the QR code when the attendee arrives.

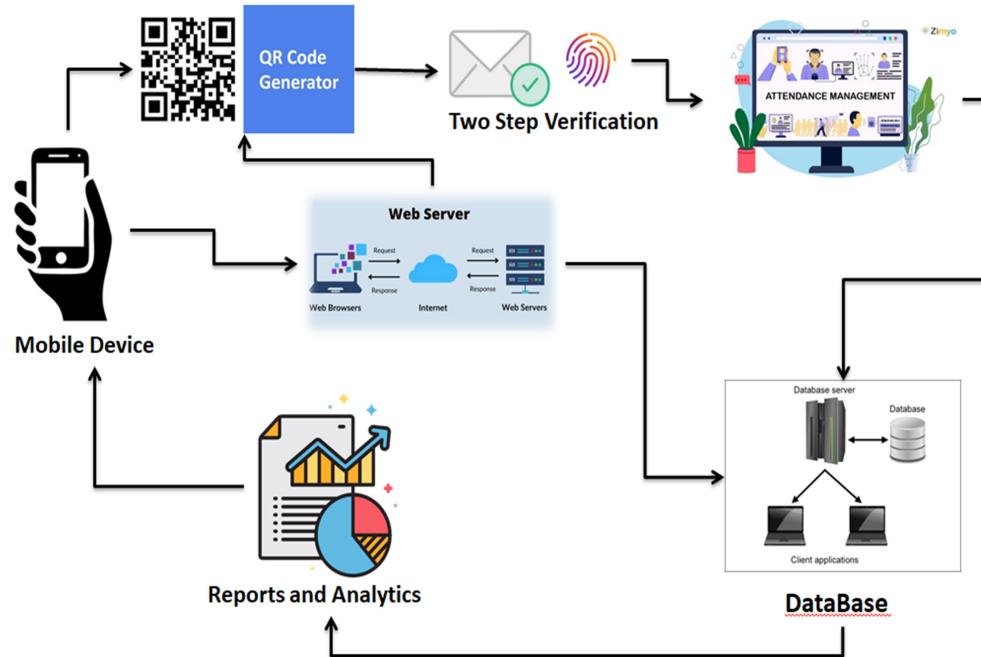
Server-side: A web application that receives the attendance data from the mobile application and processes it. The web application can be built using a web framework such as Django, Flask, or Ruby on Rails.

Database: A database to store the attendance data, such as MySQL or PostgreSQL.

Authentication and security: The system should include authentication and security features to ensure that only authorized users can access the attendance data.

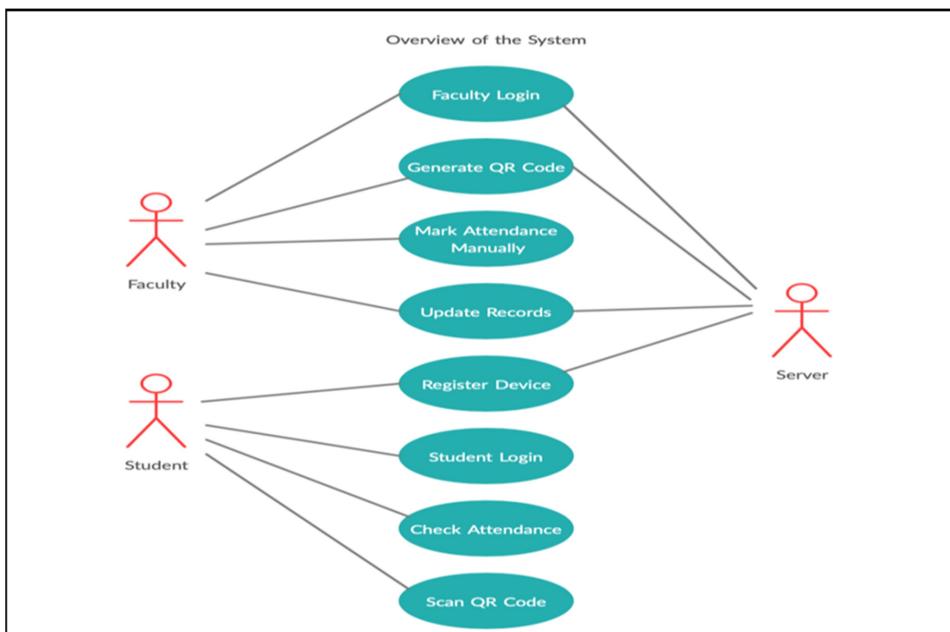
Reporting and analytics: The system can include reporting and analytics features to generate attendance reports and analyze attendance trends.

Overall, the architecture for a QR code based attendance system can be relatively simple, but it's important to ensure that the system is secure, scalable, and easy to use for both the attendees and the administrators.



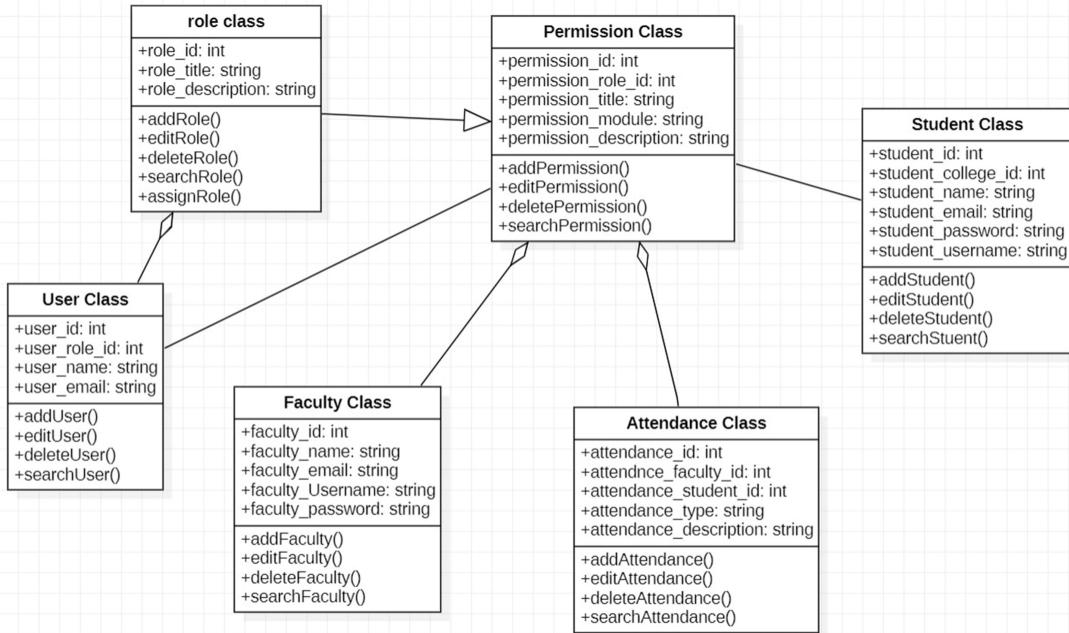
Architecture diagram(figure 9)

UseCase diagram:



UseCase diagram(figure 10)

Class Diagram:



Classdiagram(figure 11)

Result:

Thus, the system architecture, use case and class diagram created successfully.

EXPERIMENT-07

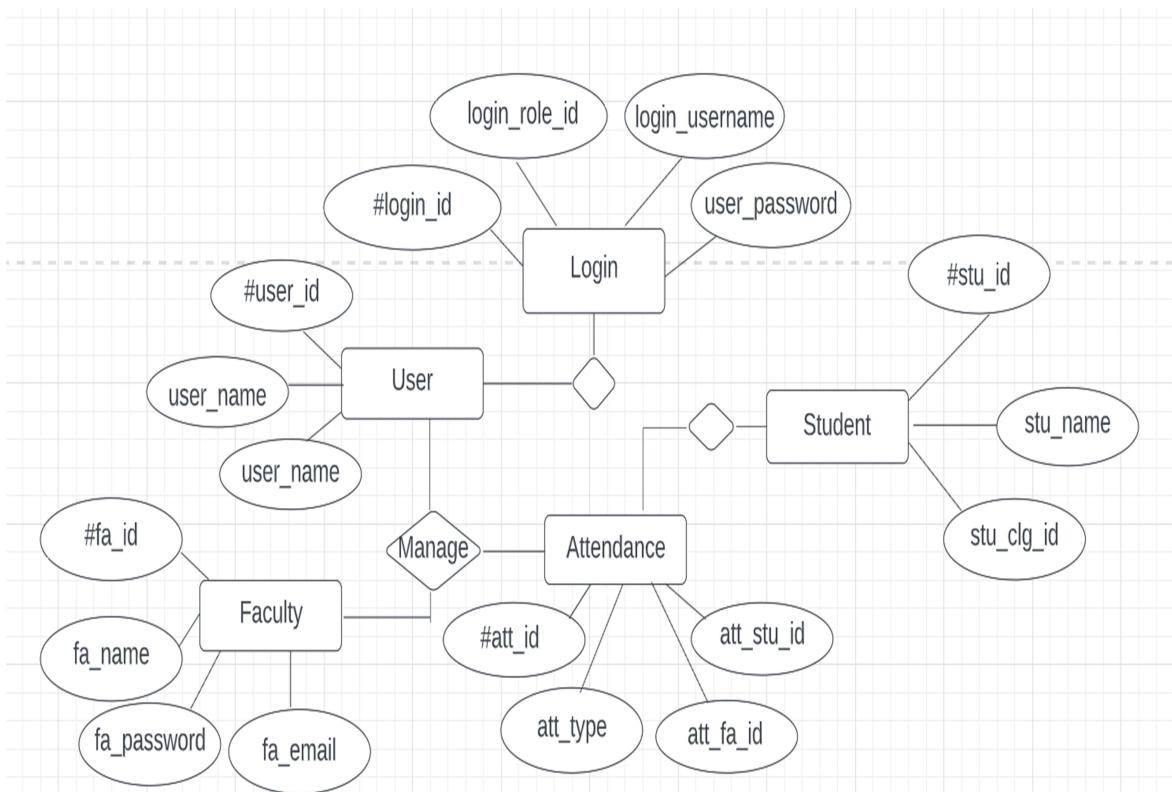
Aim

To create the Entity Relationship Diagram.

ER Diagram:

Components of the ER Diagram:

This model is based on three basic concepts: Entities, Attributes, Relationships.



Entity relationship diagram(figure 12)

ER (Entity Relationship) Diagram:

This ER (Entity Relationship) Diagram represents the model of Attendance Management System Entity. The entity-relationship diagram of Attendance Management System shows all the visual instrument of database tables and the relations between Employee, Holidays, Attendance, Login etc. It used structure data and to define the relationships between structured data groups of Attendance Management System functionalities. The main entities of the Attendance Management System are Attendance, Faculty, Student and Login.

Attendance Management System entities and their attributes:

- **Attendance Entity:** Attributes of Attendance are attendance_id, attendance_faculty_id, attendance_student_id, attendance_type.
- **Faculty Entity:** Attributes of Faculty are faculty_id, faculty_name, faculty_email, faculty_password.
- **Student Entity:** Attributes of Student are student_id, student_college_id, student_name, student_email.
- **Login Entity:** Attributes of Login are login_id, login_username, login_role_id, login_password.

Description of Attendance Management System Database:

The details of Attendance is store into the Attendance tables respective with all tables.

- Each entity (Login, Student, Faculty, Attendance) contains primary key and unique keys.
- The entity Student, Leave has binded with Attendance, Faculty entities with foreign key.
- There is one-to-one and one-to-many relationships available between Login, Attendance, Students and Faculty.

All the entities Attendance, Student, Login are normalized and reduce duplicacy of records

- We have implemented indexing on each table of Attendance Management System tables for fast query execution.

Result:

Thus, the entity relationship diagram was created successfully.

EXPERIMENT-08

Aim

To develop the data flow diagram up to level 1 for the QR based Attendance System

Project Title: QR-code Based Attendance System

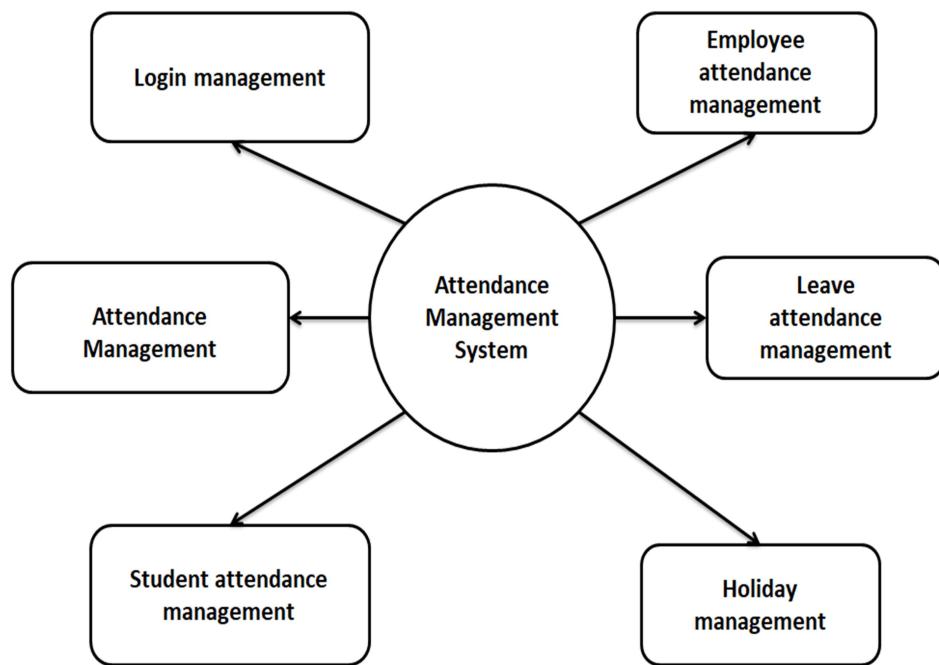
Zero Level Data Flow Diagram(0 Level DFD) Of Attendance Management System :

This is the Zero Level DFD of Attendance Management System, where we have elaborated the high level process of Attendance. It's a basic overview of the whole Attendance Management System or process being analyzed or modeled. It's designed to be an at-a-glance view of Leave, Login and Authentication showing the system as a single high-level process, with its relationship to external entities of Attendance, Employee Attendance and Student Attendance. It should be easily understood by a wide audience, including Attendance, Student Attendance and Leave. In zero level DFD of Attendance Management System, we have described the high level flow of the Attendance system.

High Level Entities and process flow of Attendance Management System:

- Managing all the Attendance
- Managing all the Employee Attendance
- Managing all the Student Attendance
- Managing all the Holidays
- Managing all the Leave
- Managing all the Login
- Managing all the Authentication

DFD Level 0(figure 13)



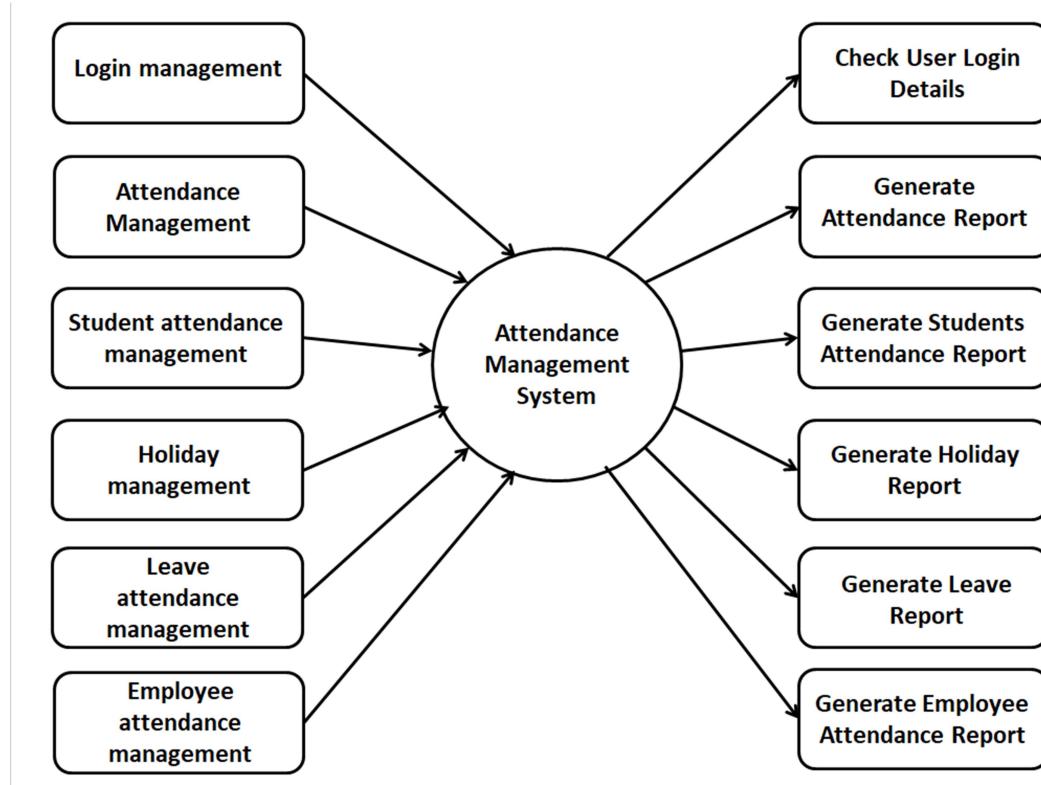
First Level Data Flow Diagram(Ist Level DFD) Of Attendance Management System :

First Level DFD (1st Level) of Attendance Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Attendance Management System system as a whole. It also identifies internal data stores of Authentication, Login, Leave, Holidays, Student Attendance that must be present in order for the Attendance system to do its job, and shows the flow of data between the various parts of Attendance, Student Attendance, Login, Authentication, Leave of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Attendance.

Main entities and output of First Level DFD (Ist Level DFD):

- Processing Attendance records and generate report of all Attendance
- Processing Employee Attendance records and generate report of all Employee Attendance
- Processing Student Attendance records and generate report of all Student Attendance
- Processing Holidays records and generate report of all Holidays
- Processing Leave records and generate report of all Leave
- Processing Login records and generate report of all Login
- Processing Authentication records and generate report of all Authentication

DFD Level 1(figure 14)



Result:

Thus, the data flow diagrams have been created for the QR based attendance system,

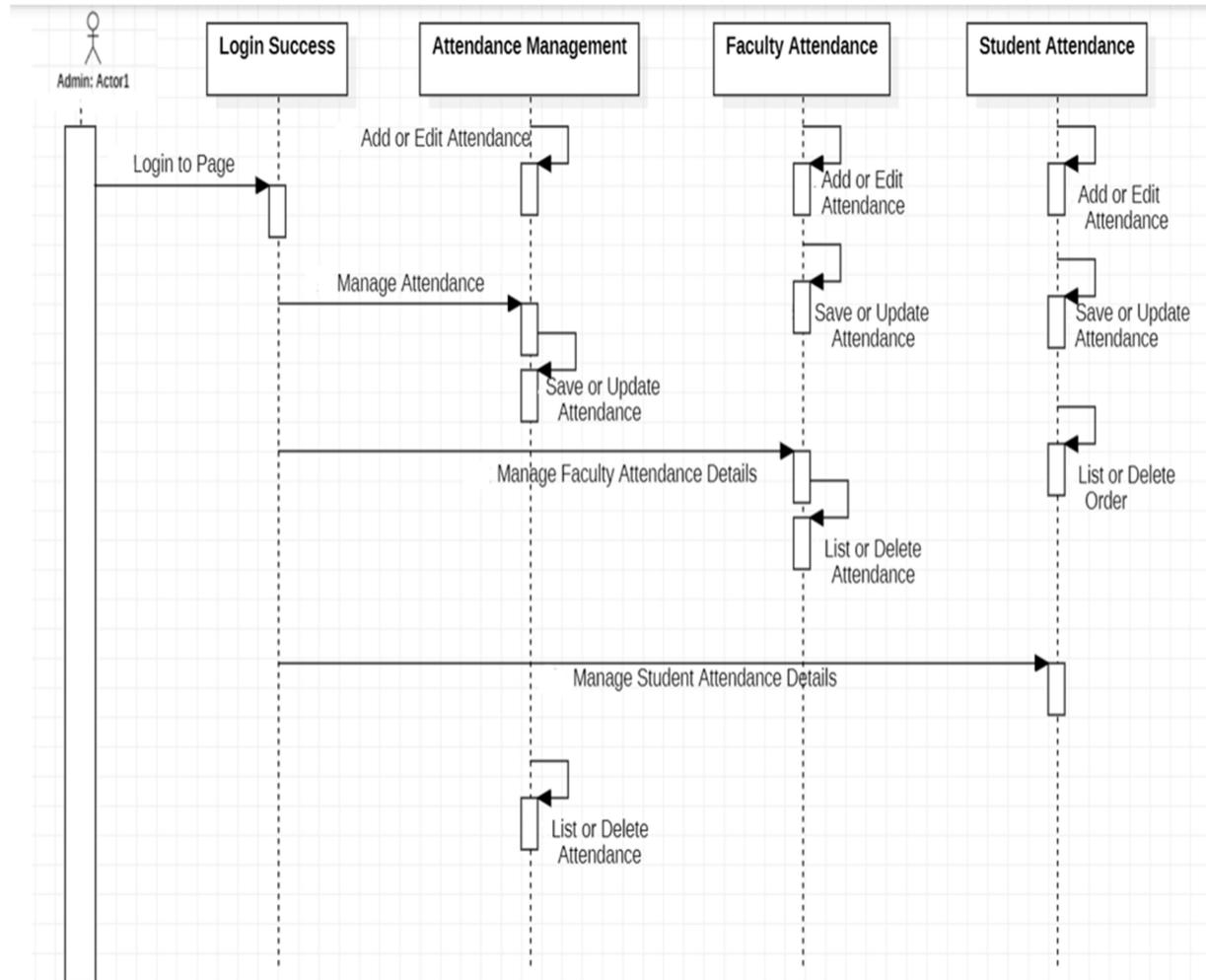
EXPERIMENT-09

Aim:

To create the sequence and collaboration diagram for the QR code based Attendance System

Project Topic: QR code based Attendance System

Sequence Diagram(figure 15)



This is the UML sequence diagram of Attendance Management System which shows the interaction between the objects of Login, Attendance, Faculty

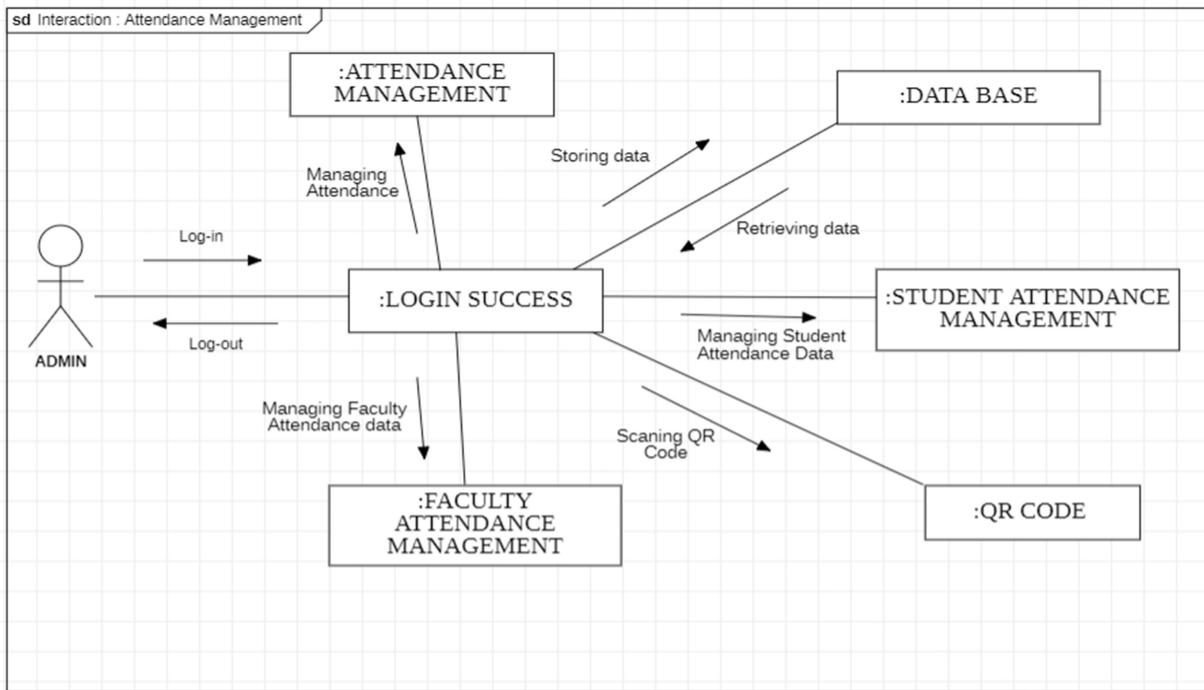
Attendance, Student Attendance. The instance of class objects involved in this UML Sequence Diagram of Attendance Management System are as follows:

- Login Object
- Attendance Object
- Student Attendance Object
- Faculty Attendance Object

Login Sequence Diagram of Attendance Management System:

This is the Login Sequence Diagram of Attendance Management System, where admin will be able to login in their account using their credentials After login user can manage all the operations on Login, Attendance, Student Attendance, Faculty Attendance. All the pages such as Attendance, Student Attendance, Employee Attendance are secure and user can access these pages after login. The diagram below helps demonstrate how the login page works in the Attendance Management System The various objects in the Student Attendance, Login, Attendance, and Faculty Attendance page interact over the course of the sequence, and user will not be able to access this page without verifying their identity.

Collaboration Diagram(figure 16)



This is the UML Collaboration diagram of Attendance Management System which shows the interaction between the objects of Login, Attendance, Faculty Attendance, Student Attendance, Database and QR code. The instance of class objects involved in this UML Sequence Diagram of Attendance Management System are as follows:

- Login Object
- Attendance Object
- Student Attendance Object
- Faculty Attendance Object
- Data base Object

- QR Code Object

This is the collaboration Diagram of Attendance Management System, where admin will be able to login in their account using their credentials After login user can manage all the operations on Login, Attendance, Student Attendance, Faculty Attendance. All the pages such as Attendance, Student Attendance, Employee Attendance are secure and user can access these pages after login. The diagram below helps demonstrate how the login page works in the Attendance Management System The various objects in the Student Attendance, Login, Attendance, and Faculty Attendance page interact over the course of the sequence, and user will not be able to access this page without verifying their identity.

Result:

Thus, the sequence and collaboration diagrams were created for the QR code based Attendance System

EXPERIMENT-10

Aim

To develop the testing framework and/or user interface framework for the QR Code Based Attendance System

Executive Summary:

Scope:

The scope of testing for the QR code based attendance system is to ensure that the application is functioning as per the business requirements and specifications. The testing should cover functional and non-functional aspects of the application, including but not limited to attendance marking, user authentication, data storage, and system performance.

Objective:

The objective of testing the QR code based attendance system is to identify any defects or issues that may exist in the application and to ensure that the system meets the requirements and expectations of the end-users. The testing should ensure that the application is reliable, secure, and user-friendly, and that it can handle a large volume of users and attendance records.

Approach:

The approach to testing the QR code based attendance system should be a structured and systematic one that covers both manual and automated testing. The testing approach should include the following:

- Test Planning: Develop a comprehensive test plan that outlines the testing strategy, scope, test objectives, and test cases.
- Test Case Development: Develop test cases that cover all aspects of the application, including functional and non-functional requirements.
- Test Execution: Execute the test cases and record the results. The testing should include both positive and negative testing scenarios.
- Defect Management: Track and manage defects found during the testing phase. The defects should be documented, prioritized, and addressed by the development team.
- Test Reporting: Generate test reports that summarize the testing results and provide an overview of the quality of the application.
- Continuous Testing: Implement continuous testing to ensure that the application remains stable and performs as expected throughout its lifecycle.

The testing phases for QR based Attendance system typically include the following:

Unit Testing: Unit testing is the first phase of testing, where individual components of the system are tested in isolation. This is done to ensure that each component is functioning as expected and to identify any defects early in the development cycle.

Integration Testing: In this phase, the individual components of the system are combined and tested together as a whole. This is done to ensure that the components are working together as expected and to identify any defects that may arise due to interactions between the components.

System Testing: System testing involves testing the entire system as a whole, including all the components and their interactions. This is done to ensure that the system meets the functional and non-functional requirements and to identify any defects that may have been missed in the previous testing phases.

User Acceptance Testing: User acceptance testing (UAT) is conducted to ensure that the system meets the user requirements and is easy to use. UAT is typically conducted by end-users or stakeholders who are familiar with the system requirements.

Performance Testing: Performance testing is done to ensure that the system meets the performance requirements, such as response time, throughput, and resource utilization. This is done to ensure that the system can handle the expected workload and can scale to meet future demand.

Security Testing: Security testing is done to ensure that the system is secure and protected from unauthorized access or attacks. This includes testing the system's access controls, encryption, and vulnerability assessments.

Regression Testing: Regression testing is conducted to ensure that the changes made to the system during the development cycle have not impacted the existing functionality of the system. Overall, these testing phases are conducted in a systematic manner to ensure that the QR code based attendance system system is thoroughly tested and meets the requirements and expectations of the stakeholders.

In summary, the testing approach for the QR code based attendance system should be comprehensive and cover all aspects of the application. The objective of testing is to ensure that the application is reliable, secure, and user-friendly, and that it meets the requirements and expectations of the end-users.

Testing Approach

The testing approach for a QR code based attendance system app would involve the following steps:

Requirement Analysis

The first step in testing the QR code based attendance system would be to analyze the requirements of the app to ensure that all the features and functionalities are identified and understood.

Test Planning

After analyzing the requirements, a test plan should be created that outlines the test objectives, scope, approach, and test schedule. The test plan should also identify the testing tools and techniques that will be used.

Test Design

In this phase, test cases are designed to validate the functionality and usability of the app. Test cases should be designed to test all the features of the app, including basic functionality, error handling, performance, and security.

Test Execution

Once the test cases have been designed, they should be executed to ensure that the app functions as expected. The test results should be

documented, and defects should be logged in a defect tracking system.

Test Reporting

After executing the test cases, a test report should be created that summarizes the results of the testing. The report should include information on the test coverage, test results, and any defects that were found.

Regression Testing

Regression testing should be performed after any changes or updates are made to the app to ensure that existing functionalities are not impacted and that new functionalities are working as expected.

User Acceptance Testing

After the testing has been completed, the app should be tested by end-users to ensure that it meets their needs and is easy to use. Overall, the testing approach for a vehicle maintenance system app should be comprehensive, covering all aspects of the app's functionality, performance, usability, security, and integration with other systems. The testing should be conducted in multiple phases and should involve the use of various testing tools and techniques to ensure that the app is thoroughly tested.

Testing Environment

The testing environment for a QR code based attendance system should be representative of the production environment in which the system will operate. This includes hardware, software, and network configurations that are similar to those in the production environment. Here are some key components that should be included in the testing environment:

Hardware: The hardware should be similar to the production environment in terms of the type of computers, servers, and mobile devices that will be used by the end-users. This includes ensuring that the hardware has the required processing power, memory, and storage capacity to handle the application.

Software: The software used in the testing environment should be the same as that used in the production environment. This includes the operating system, database management system, web server, and any other software components used by the application.

Network: The network infrastructure used in the testing environment should be similar to the production environment in terms of the network topology, protocols, and bandwidth. This is important to ensure that the application performs well under realistic network conditions.

Test Data: The testing environment should have a representative sample of the data that will be used in the production environment. This includes customer data, transaction data, and any other data that is required to test the functionality of the application.

Test Tools: The testing environment should have all the required testing tools and software installed, including testing frameworks, test automation tools, and debugging tools.

Security: The testing environment should be secured to ensure that the application is protected from unauthorized access or attacks. This includes securing the network, servers, and any other components that are part of the application.

Overall, the testing environment should be well-designed and maintained to ensure that the testing is effective and that the application is thoroughly tested before it is released into the production environment.

Scope of Testing for QR Code Based Attendance System

The scope of testing for the QR code based attendance system includes both functional and non-functional aspects of the application. The testing aims to ensure that the system meets the business

requirements and specifications, and that it is reliable, secure, and user-friendly.

Functional Testing:

The functional testing will cover all modules of the application, including user authentication, QR code scanning, attendance marking, and data storage. The testing will include both positive and negative scenarios to ensure that the application works as intended.

Automation will cover all critical path test cases, and regression testing will be performed to ensure that any changes to the application do not affect its existing functionality.

Non-Functional Testing:

The non-functional testing will cover all non-functional requirements, including performance, security, usability, and scalability.

Performance testing will be conducted to ensure that the application can handle a large volume of users and attendance records. Security testing will be performed to identify and address any vulnerabilities in the system. Usability testing will be conducted to ensure that the application is user-friendly and easy to navigate. Scalability testing will be conducted to ensure that the system can handle an increase in user and attendance volume.

In summary, the scope of testing for the QR code based attendance system includes both functional and non-functional aspects of the application. The testing will cover all modules and non-functional requirements to ensure that

the system meets the business requirements and specifications and is reliable, secure, and user-friendly.

<u>Types of Testing</u>	<u>Methodology</u>	<u>Tools Required</u>
Functional Testing	<ul style="list-style-type: none"> Identify all the features of the QR code-based attendance system. Develop test cases for each feature, covering both positive and negative scenarios. Execute the test cases and verify if all features are working as intended. 	<ul style="list-style-type: none"> Selenium WebDriver for web-based systems Appium for mobile-based systems TestRail for test case management
Integration Testing	<ul style="list-style-type: none"> Identify all the external systems that the QR code-based attendance system needs to integrate with. Develop test cases to verify the integration of the attendance system with each external system. Execute the test cases and verify if the integration works as intended. 	<ul style="list-style-type: none"> SoapUI for testing web services Postman for testing RESTful APIs Jenkins for continuous integration and testing

<u>Types of Testing</u>	<u>Methodology</u>	<u>Tools Required</u>
Performance Testing	<ul style="list-style-type: none"> Identify the expected load on the system. Develop test scenarios that simulate the expected load. Execute the test scenarios and measure the system's performance, including response time, throughput, and resource utilization. 	<ul style="list-style-type: none"> JMeter for load testing Gatling for stress testing New Relic for application monitoring
Usability Testing	<ul style="list-style-type: none"> Identify the target users of the attendance system. Develop test cases to evaluate the system's usability, including the interface, navigation, and user experience. Conduct user testing and gather feedback from users to evaluate the system's usability. 	<ul style="list-style-type: none"> UserTesting.com for remote user testing Optimal Workshop for user research and testing Usabilla for collecting user feedback
Security Testing	<ul style="list-style-type: none"> Identify potential security threats and vulnerabilities. Develop test cases to verify the system's security features, such as authentication, authorization, and encryption. Conduct penetration testing to identify any weaknesses in the system's security. 	<p>OWASP ZAP for vulnerability scanning Nessus for network scanning Burp Suite for web application security testing</p>

Result:

Thus, the testing framework/user interface framework has been created for the QR
Code Based Attendance System

EXPERIMENT-11

Aim

To develop the test cases manual for the QR code based Attendance system

Functional Test Cases

<u>Test ID</u>	<u>Test Scenario</u>	<u>Test Case</u>	<u>Execution Steps</u>	<u>Expected Outcome</u>	<u>Actual Outcome</u>	<u>Status</u>	<u>Remarks</u>
1	Scan a valid QR code	QR-001	<ol style="list-style-type: none">1. Obtain a valid QR code associated with a registered user in the attendance system.2. Launch the attendance app and navigate to the scan QR code feature.3. Scan the QR code using the device's camera.	The system should create an attendance record for the corresponding user and date in the system. A confirmation message or visual indicator should be displayed to confirm the scan was successful.	confirmation message or visual indicator should be displayed successfully	Pass	The QR code was successfully scanned
2	Scan an invalid QR code	QR-002	<ol style="list-style-type: none">1. Obtain an invalid QR code that is not associated with a registered user in the	The system should display an appropriate error message and prevent the creation of an attendance	The system should display an appropriate error message and prevent the creation of an	Pass	The QR code appears to be invalid. Please try scanning a valid QR code

			attendance system.	record.	attendance record was created successfully		
			2. Launch the attendance app and navigate to the scan QR code feature. 3. Scan the QR code using the device's camera.				
3	Scan a damaged QR code	QR-003	1. Obtain a QR code that is physically damaged or distorted. 2. Launch the attendance app and navigate to the scan QR code feature. 3. Scan the QR code using the device's camera.	The system should detect that the QR code is damaged and display an appropriate error message.	The system detected that the QR code is damaged and display an appropriate error message.	Failed	Sorry, but the QR code is damaged and cannot be scanned. Please obtain a new QR code.
4	Scan a QR code using different devices	QR-004	1. Obtain a valid QR code associated with a registered user in the attendance system. 2. Launch	The system should be able to scan the QR code accurately and create an attendance record for the corresponding user and date in the system.	The system scanned the QR code accurately and create an attendance record for the corresponding user and date in the	Pass	The QR code was successfully scanned using different devices.

			<p>the attendance app on different devices such as smartphones, tablets, and laptops.</p> <p>3. Navigate to the scan QR code feature.</p> <p>4. Scan the QR code using the device's camera.</p>		system		
5	Verify attendance data accuracy	QR-005	<p>1. Create attendance records manually or using other methods.</p> <p>2. Obtain valid QR codes associated with the registered users.</p> <p>3. Scan the QR codes using the attendance app.</p>	<p>The attendance data recorded in the system should match the attendance data recorded manually or using other methods.</p>	<p>The attendance data recorded in the system matched the attendance data recorded manually .</p>	Pass	<p>The attendance data appears to be accurate and matches the information from the QR code.</p>

Non-Functional Test Cases

<u>Test ID</u>	<u>Test Scenario</u>	<u>Test Case</u>	<u>Execution Steps</u>	<u>Expected Outcome</u>	<u>Actual Outcome</u>	<u>Status</u>	<u>Remarks</u>
1	Performance Testing	PF-001	1. Test the attendance system with a large number of concurrent users. 2. Monitor the system's response time, CPU and memory usage, and network traffic.	The system should be able to handle the load of a large number of concurrent users without significant performance degradation.	The system able to handle the load of a large number of concurrent users without significant performance degradation.	Pass	The test results indicate that the system can handle expected user load and workloads within acceptable performance limits. However, there are some areas where further optimization could be made to improve performance and ensure a better user experience.
2	Usability Testing	UT-001	1. Ask a sample of users to perform common tasks in the attendance system, such as scanning a QR code, viewing attendance records, and modifying user information. 2. Collect feedback from the users about the ease of use and intuitiveness of the system.	The system should be easy to use and intuitive, with clear instructions and visual cues for common tasks. Users should be able to navigate the system without difficulty.	The system is easy to use and intuitive, with clear instructions and visual cues for common tasks. Users is able to navigate the system without difficulty or confusion.	Pass	The test results show that the software is user-friendly and easy to navigate, and users were able to complete tasks efficiently and effectively. However, there are some areas where further improvements could be made to enhance the user experience and make the software more intuitive.
3	Security	ST-001	1. Test the attendance system's security	The system should be secure and able to prevent	The system is secure and able to prevent	Pass	The test results show that the system is secure and meets the security requirements.

<u>Test ID</u>	<u>Test Scenario</u>	<u>Test Case</u>	<u>Execution Steps</u>	<u>Expected Outcome</u>	<u>Actual Outcome</u>	<u>Status</u>	<u>Remarks</u>
	Testing		<p>measures, such as authentication, encryption, and access control.</p> <p>2. Attempt to bypass or circumvent the security measures, such as by using a fake or modified QR code or attempting to access another user's attendance record.</p>	unauthorized access or tampering of attendance records.	unauthorized access or tampering of attendance records.		necessary security requirements. However, there are some areas where further improvements could be made to enhance security and mitigate potential risks.
4	Compatibility Testing	CT-001	<p>1. Test the attendance system on different devices, operating systems, and web browsers.</p> <p>2. Ensure that the system works properly and is fully functional on all tested configurations.</p>	The system should be compatible with a wide range of devices, operating systems, and web browsers, and should function properly on all tested configurations.	The system is compatible with a wide range of devices, operating systems, and web browsers, and should function properly on all tested configurations.	Pass	The test results show that the system is compatible with a range of different platforms and operating systems, which will ensure a broader user base. However, there are some areas where further compatibility testing could be made to ensure full compatibility across all platforms.
5	Accessibility Testing	AT-001	<p>1. Test the attendance system's accessibility features, such as screen readers and keyboard</p>	The system should be accessible to users with disabilities or impairments, with clear and concise instructions, proper labeling of	The system is accessible to users with disabilities or impairments, with clear and concise instructions, proper labeling of	Pass	The test results show that the system is accessible to users with different abilities and meets the necessary accessibility requirements. However, there are some areas where further improvement could be made to enhance accessibility.

<u>Test ID</u>	<u>Test Scenario</u>	<u>Test Case</u>	<u>Execution Steps</u>	<u>Expected Outcome</u>	<u>Actual Outcome</u>	<u>Status</u>	<u>Remarks</u>
			<p>navigation.</p> <p>2. Ensure that users with disabilities or impairments can use the system effectively.</p>	controls and links, and support for assistive technologies.	controls and links, and support for assistive technologies.		and ensure the software is fully inclusive.

Result:

Thus, the test case manual has been created for the QR code Based Attendance System.

EXPERIMENT-12

Aim

To prepare the manual test case report for the QR code Based Attendance management System

Summary

Functional testing of a QR code-based attendance management system:

Functional testing of a QR code-based attendance management system would involve verifying that the system functions as expected in terms of the functional requirements. Here are some test scenarios and test cases for functional testing of a QR code-based attendance management system:

Test scenario: Generating QR codes

- Test case 1: Verify that a unique QR code is generated for each student/employee.
- Test case 2: Verify that the QR code contains the necessary information such as name, student/employee ID, date, and time.

Test scenario: Scanning QR codes

- Test case 1: Verify that the system can scan QR codes from different angles and distances.
- Test case 2: Verify that the system can recognize the QR code and retrieve the necessary information.

Test scenario: Marking attendance

- Test case 1: Verify that the system marks attendance only when a valid QR code is scanned.
- Test case 2: Verify that the system records the date and time of attendance for each student/employee.

Test scenario: Viewing attendance records

- Test case 1: Verify that the system displays the correct attendance record for each student/employee.
- Test case 2: Verify that the system can filter attendance records by date, student/employee, and class/group.

Test scenario: Reporting

- Test case 1: Verify that the system generates accurate attendance reports for individual students/employees and groups/classes.
- Test case 2: Verify that the system can export attendance reports in different formats such as PDF, Excel, and CSV.

Test scenario: Security

- Test case 1: Verify that the system restricts access to attendance records and reports to authorized users only.
- Test case 2: Verify that the system encrypts sensitive information such as student/employee data and attendance records.

By testing these scenarios and test cases, you can ensure that the QR code-based attendance management system meets the functional requirements and performs its intended functions accurately and reliably.

Non-functional testing of a QR code-based attendance management system

Non-functional testing of a QR code-based attendance management system would involve verifying the system's performance, reliability, usability, security, and compatibility with other systems. Here are some test scenarios and test cases for non-functional testing of a QR code-based attendance management system:

Test scenario: Performance

Test case 1: Verify that the system can handle a large number of students/employees and attendance records without slowing down or crashing.

Test case 2: Verify that the system responds quickly when generating QR codes, scanning codes, and generating reports.

Test scenario: Reliability

Test case 1: Verify that the system is available and accessible to authorized users at all times.

Test case 2: Verify that the system is resilient to system failures such as server crashes or network outages.

Test scenario: Usability

Test case 1: Verify that the system is user-friendly and intuitive, with clear instructions and prompts for users.

Test case 2: Verify that the system is accessible to users with different levels of technical expertise and physical abilities.

Test scenario: Security

Test case 1: Verify that the system has adequate measures in place to protect student/employee data and attendance records from unauthorized access and data breaches.

Test case 2: Verify that the system enforces strong authentication and password policies to prevent unauthorized access.

Test scenario: Compatibility

Test case 1: Verify that the system is compatible with different operating systems, browsers, and devices.

Test case 2: Verify that the system integrates seamlessly with other systems and applications used by the school or organization.

By testing these scenarios and test cases, you can ensure that the QR code-based attendance management system is reliable, efficient, secure, user-friendly, and compatible with other systems and applications used by the school or organization.

Obstacles in Functional testing

Complex business logic: The attendance management system can have complex business logic, such as rules for calculating attendance, handling exceptions, and generating reports. It can be challenging to ensure that the system correctly implements the business logic and meets the requirements.

Solution:

The testing team should collaborate closely with the development team and subject matter experts to understand and document the business logic in detail. The team can then use the information to design and execute test scenarios and test cases that cover all possible scenarios and edge cases. The use of automation testing can also help to streamline testing for complex business logic.

Integration with other systems: The attendance management system can integrate with other systems, such as payroll or HR systems. It can be challenging to test the integration points and ensure that data flows correctly between the systems.

Solution:

The testing team should work closely with the development team and third-party system vendors to understand the integration points and design and execute test cases that validate data flows between systems. The use of mock services and APIs can help to simulate the interactions between systems and test the integration points in isolation.

User interface (UI) design: The attendance management system can have a complex UI design that includes multiple forms, screens, and workflows. It can be challenging to test the UI design and ensure that the system is user-friendly and intuitive.

Solution:

The testing team should collaborate with the UX/UI design team to understand the design principles and user requirements. The team can then design and execute test cases that cover the usability, accessibility, and user experience aspects of the system. The use of manual and automated UI testing tools can help to streamline UI testing and identify issues early in the development cycle.

Data accuracy and integrity: The attendance management system can rely on accurate and reliable data to function correctly. It can be challenging to ensure that the system correctly handles data input, validation, storage, and retrieval.

Solution:

The testing team should design and execute test cases that cover data input, validation, storage, and retrieval. The team should also validate that the data flows correctly between different system components and that the system enforces data integrity constraints. The use of data generation and data masking tools can help to generate test data that covers different scenarios and usage patterns.

System performance: The attendance management system can be used by a large number of users, and it can be challenging to ensure that the system meets performance requirements, such as response time, scalability, and availability.

Solution:

The testing team should design and execute performance testing scenarios that simulate real-world usage patterns and user loads. The team should also use performance monitoring tools to identify bottlenecks and issues that affect system performance. The use of performance testing tools and techniques such as load testing, stress testing, and scalability testing can help to identify performance issues early in the development cycle.

By identifying and overcoming these obstacles, the testing team can ensure that the QR code-based attendance management system meets functional requirements and user expectations. The testing team should also work closely with the development team and other stakeholders to identify and address issues early in the development cycle and ensure that the system is delivered on time and within budget.

Obstacles in Non -Functional testing

Limited access to production environment: Non-functional testing requires a replica of the production environment, which can be challenging to set up and access due to security and access restrictions.

Solution:

One solution could be to create a test environment that replicates the production environment as closely as possible. The test environment can be set up with similar hardware, software, and network configurations as the production environment to simulate the real-world scenario. Another solution is to use virtualization or cloud-based services to replicate the production environment in a controlled and secure manner.

Test data availability: Non-functional testing requires a large volume of test data, and it can be challenging to generate or obtain sufficient data to test the system's performance or reliability accurately.

Solution:

To overcome this obstacle, you can generate realistic and representative test data that covers different scenarios and usage patterns. Test data can also be obtained from production data or by using data generation tools. If the test data is sensitive or confidential, it can be anonymized or masked to protect the data privacy.

Performance bottlenecks: Identifying the performance bottlenecks can be challenging as there could be multiple factors affecting the system's performance, such as network latency, system load, or database performance.

Solution:

To identify performance bottlenecks, you can use specialized testing tools such as load testing, stress testing, and performance monitoring tools. These tools can

simulate user load and measure the system's response time, throughput, and resource utilization. You can also conduct testing on different hardware configurations to identify performance issues related to hardware.

Integration challenges: If the QR code-based attendance management system integrates with other systems, it can be challenging to test its compatibility with those systems, as there could be compatibility issues with different operating systems, devices, or databases.

Solution:

To overcome integration challenges, you can use integration testing tools and techniques to test the system's compatibility with other systems and applications. Integration testing can be performed by simulating the interactions between the QR code-based attendance management system and other systems or by using real-world data. Integration testing can help identify compatibility issues related to different operating systems, devices, or databases.

Testing tools and techniques: Non-functional testing requires specialized tools and techniques to measure performance, security, or reliability. These tools can be expensive, and the testing team may require specialized training to use them effectively.

Solution:

To overcome this obstacle, you can invest in specialized testing tools and techniques that are suitable for non-functional testing. You can also provide training and support to the testing team to help them use the testing tools and techniques effectively. You can also collaborate with external vendors or consultants who have expertise in non-functional testing to supplement the in-house team's skills and knowledge.

By identifying and overcoming these obstacles, you can ensure that the QR code-based attendance management system meets the required performance, reliability, usability, security, and compatibility standards.

Testing Process

Functional Test-Cases

Test Case Execution: This step is to execute the designed functional test cases manually, step-by-step, to verify if the actual output of the application matches the expected output specified in the test case. During test case execution, the tester verifies all the functionalities of the software application, including the input validations, business logic, user interface, error messages, and other functional requirements.

Defect Reporting: During the test case execution, if the actual output does not match the expected output, the tester identifies the issue and reports it as a defect in a defect tracking tool. The defect report typically includes the steps to reproduce the issue, the expected behavior, the actual behavior, and any relevant screenshots or logs.

Defect Verification: After the defects are reported, the development team investigates the issue and fixes the defect. The tester then verifies the defect fix to ensure that the expected behavior is achieved.

Non-Functional Test-Cases

Performance Testing: In performance testing, test cases are designed to measure the speed, scalability, and stability of the software application under different load conditions. The test cases typically include scenarios such as measuring the response time of the application, testing the application under heavy load, and measuring the system resources used by the application under different load conditions.

Compatibility Testing: In compatibility testing, test cases are designed to ensure that the software application works correctly with different hardware, software, and operating system configurations. The test cases typically include scenarios such as testing the application on different browsers, testing the application on different devices, and testing the application on different operating systems.

Integration Testing: Integration testing is a type of testing that is performed to verify that different components of the software application work together correctly. In manual integration testing, the tester typically follows a test plan that outlines the steps needed to test the integration points between different components. The tester will perform the tests and record the results, looking for any issues or bugs that may arise when the components are integrated together.

User Acceptance Testing (UAT): User Acceptance Testing is a type of testing that is performed to ensure that the software application meets the user's requirements and expectations. In manual UAT testing, the tester typically follows a test plan that outlines the user scenarios to be tested. The tester will perform the tests and record the results, looking for any issues or bugs that may arise when the user scenarios are executed. The goal of UAT testing is to ensure that the software application is ready for release to the end-users.

Recovery Testing: Recovery Testing is a type of testing that is performed to verify that the software application can recover from various failures or crashes. In manual recovery testing, the tester typically follows a test plan that outlines the failure scenarios to be tested. The tester will perform the tests and record the results, looking for any issues or bugs that may arise when the application is recovering from a failure or crash. The goal of recovery testing is to ensure that the software application can recover gracefully and continue to function correctly after a failure or crash.

Category	Progress Against Plan	Status	Software
Functional Testing	Greeen	Completed	—
Log-in	100%	Completed	—
QR Code Scanning	100%	Completed	—
Attendance Recording	100%	Completed	—
Reporting	100%	Completed	—
Non-Functional Testing	Amber	In-Progress	—
Performance Testing	90%	In-Progress	Selenium
Compatibility Testing	100%	Completed	Selenium
Integration Testing	100%	Completed	SoapUI
User Acceptance Testing	75%	In-Progress	—
Security Testing	75%	In-Progress	OWASP ZAP
Recovery Testing	90%	In-Progress	OWASP ZAP

Testing Software Report:-

Selenium: (Functional Test Case Testing)-figure 17

The screenshot shows the Selenium IDE interface with the following details:

- Session Name:** Executing - Vehicare*
- URL:** http://localhost:50991/
- Test Cases:** A table listing four commands:
 - 1. ✓ open /
 - 2. ✓ set window size 1078x816
 - 3. click css=flutter-view
 - 4. close
- Tool Buttons:** Top right icons for play, stop, refresh, and save.
- Input Fields:** Command, Target, Value, and Description fields.
- Status Bar:** Runs: 0 Failures: 0
- Log Tab:** Active tab.

OWASP ZAP : (Security Testing)-figure 18

The screenshot shows the OWASP ZAP interface with the following details:

- Title:** Untitled Session - Testing Vehicare - OWASP ZAP 2.12.0
- Menu Bar:** File, Edit, View, Analyse, Report, Tools, Import, Export, Online, Help
- Toolbar:** Standard Mode, Quick Start, Request, Response, Requester, +
- Main Area:** "Automated Scan" section.
 - Text: "This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'."
 - Text: "Please be aware that you should only attack applications that you have been specifically been given permission to test."
 - Form:** URL to attack: https://console.firebaseio.google.com/u/0/project/vehiclemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A
 - Options:** Use traditional spider: Use ajax spider: with Firefox Headless
 - Buttons:** Attack, Stop
 - Progress:** Attack complete - see the Alerts tab for details of any issues found
- Bottom Status:** Alerts: 1 5 4 5, Main Proxy: localhost 8080, Current Scans: 0 0 0 0 0 0 0 0 0 0

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	1 (6.7%)	0 (0.0%)	1 (6.7%)
	Medium	0 (0.0%)	4 (26.7%)	0 (0.0%)	1 (6.7%)	5 (33.3%)
	Low	0 (0.0%)	2 (13.3%)	2 (13.3%)	0 (0.0%)	4 (26.7%)
	Informational	0 (0.0%)	1 (6.7%)	2 (13.3%)	2 (13.3%)	5 (33.3%)
	Total	0 (0.0%)	7 (46.7%)	5 (33.3%)	3 (20.0%)	15 (100%)

Alert counts by site and risk

This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

Site	Risk				Informational al)
	High (= High)	Medium (>= Medium)	Low (>= Informational (>= Low))		
	1 (1)	5 (6)	4 (10)	5 (15)	
https://console.firebaseio.go gle.com					

Alerts

Risk=High, Confidence=Medium (1)

<https://console.firebaseio.google.com> (1)

Cross Site Scripting (Reflected) (1)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=%27%3Balert%281%29%3B%27&utm_medium=email&utm_campaign=welcome_2021_CTA_A

Risk=Medium, Confidence=High (4)

<https://console.firebaseio.google.com> (4)

CSP: Wildcard Directive (1)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

CSP: script-src unsafe-inline (1)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

CSP: style-src unsafe-inline (1)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

Content Security Policy (CSP) Header Not Set (1)

- ▶ GET <https://console.firebaseio.google.com/sitemap.xml>

Risk=Low, Confidence=Medium (2)

<https://console.firebaseio.google.com> (2)

[Cookie without SameSite Attribute \(1\)](#)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

[Cross-Domain JavaScript Source File Inclusion \(1\)](#)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

Risk=Informational, Confidence=High (1)

<https://console.firebaseio.google.com> (1)

[Content Security Policy \(CSP\) Report-Only Header Found \(1\)](#)

- ▶ GET <https://console.firebaseio.google.com/sitemap.xml>

Risk=Informational, Confidence=Medium (2)

<https://console.firebaseio.google.com> (2)

[Modern Web Application \(1\)](#)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

[User Agent Fuzzer \(1\)](#)

- ▶ GET https://console.firebaseio.google.com/u/0/project/vechilemaintenance/overview?utm_source=welcome&utm_medium=email&utm_campaign=welcome_2021_CTA_A

Risk=Informational, Confidence=Low (2)

Alert type	Risk	Count
Cross Site Scripting (Reflected)	High	3 (20.0%)
Absence of Anti-CSRF Tokens	Medium	1 (6.7%)
CSP: Wildcard Directive	Medium	4 (26.7%)
CSP: script-src unsafe-inline	Medium	2 (13.3%)
CSP: style-src unsafe-inline	Medium	4 (26.7%)
Content Security Policy (CSP) Header Not Set	Medium	2 (13.3%)
CSP: Notices	Low	6 (40.0%)
Cookie without SameSite Attribute	Low	1 (6.7%)
Cross-Domain JavaScript Source File Inclusion	Low	2 (13.3%)
Strict-Transport-Security Header Not Set	Low	3 (20.0%)
Content Security Policy (CSP) Report-Only Header Found	Informational	2 (13.3%)
Information Disclosure - Suspicious Comments	Informational	4 (26.7%)
Modern Web Application	Informational	2 (13.3%)
User Agent Fuzzer	Informational	12 (80.0%)
User Controllable HTML Element Attribute (Potential XSS)	Informational	2 (13.3%)
Total		15

Result:

**Thus, the test case report has been created for the QR code Based
Attendance management System**

EXPERIMENT-13

Aim

To provide the details of architectural design/framework/implementation

CODE:

LOGIN CODE

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:qr_attendence_system/services/constants.dart';

class Authentication {
  createUser(String name, String email, String password) async {
    UserCredential user;
    try {
      user = await FirebaseAuth.createUserWithEmailAndPassword(
          email: email, password: password);
    } catch (e) {
      print(e);
      return e;
    }
    if (user.additionalUserInfo!.isNewUser) {
```

```
        userref.doc(user.user!.uid.toString()).set({
            'name': name,
            'isAdmin': false,
            'joinedclasses': {},
        });
    }
    return true;
}

loginUser(String email, String password) async {
    UserCredential user;
    try {
        user = await firebaseAuth.signInWithEmailAndPassword(
            email: email, password: password);
    } catch (e) {
        print(e);
        return e;
    }
    // print(user.user!.uid.toString());

    return user;
    // print(user);
}

checkadmin(String userid) async {
    DocumentSnapshot snapshot = await userref.doc(userid).get();
}
```

```
        bool isAdmin = snapshot.get('isAdmin');

        return isAdmin;
    }

}
```

Other Services

```
import 'package:cloud_firestore/cloud_firestore.dart';

import 'package:qr_attendence_system/models/classmodel.dart';
import 'package:qr_attendence_system/services/constants.dart';
import 'package:qr_attendence_system/services/userservices.dart';

class Adminservices {
    addnewclass(String classname, String classdescription) async {
        try {
            DocumentReference classid = await classref.add({
                'classname': classname,
                'classdescription': classdescription,
                'joinedusers': {},
            });
            return classid;
        } catch (e) {
            print(e.toString());
            return e;
        }
    }
}
```

```
Stream<DocumentSnapshot> getStreamofjoinedusers({required clas  
skey}) {  
    Stream<DocumentSnapshot> stream = classref.doc(classkey).snapsho  
ts();  
    return stream;  
}  
  
Future getallclasses() async {  
    QuerySnapshot querySnapshot = await classref.get();  
    List<Classmodel> classmodellist = [];  
  
    for (DocumentSnapshot doc in querySnapshot.docs) {  
        classmodellist.add(  
            Classmodel(  
                classkey: doc.id.toString(),  
                classname: doc.get('classname').toString(),  
                classdescription: doc.get('classdescription'),  
                joinedusers: doc.get('joinedusers'),  
            ),  
        );  
    }  
    return classmodellist;  
}
```

```
deleteclass( {required String classid}) async {
    DocumentSnapshot snapshot = await classref.doc(classid).get();
    Userservices userservices = Userservices();
    Map joinedusers = snapshot.get('joinedusers');
    for (String userids in joinedusers.keys) {
        userservices.removefromclass(classid: classid, userid: userids);
    }
    await classref.doc(classid).delete();
    print('deleted');
}
}
```

Flutter:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>CFBundleDevelopmentRegion</key>
    <string>en</string>
    <key>CFBundleExecutable</key>
    <string>App</string>
    <key>CFBundleIdentifier</key>
    <string>io.flutter.flutter.app</string>
    <key>CFBundleInfoDictionaryVersion</key>
```

```
<string>6.0</string>
<key>CFBundleName</key>
<string>App</string>
<key>CFBundlePackageType</key>
<string>FMWK</string>
<key>CFBundleShortVersionString</key>
<string>1.0</string>
<key>CFBundleSignature</key>
<string>????</string>
<key>CFBundleVersion</key>
<string>1.0</string>
<key>MinimumOSVersion</key>
<string>11.0</string>
</dict>
</plist>
<?xml version="1.0" encoding="UTF-8"?>
<Workspace
    version = "1.0">
    <FileRef
        location = "self:">
    </FileRef>
</Workspace>

<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>PreviewsEnabled</key>
    <false/>
</dict>
</plist>
# Miscellaneous
*.class
*.log
*.pyc
*.swp
.DS_Store
.atom/
.buildlog/
.history
.svn/
migrate_working_dir/
# IntelliJ related
*.iml
*.ipr
*.iws
.idea/
```

```
# The .vscode folder contains launch configuration and tasks you
configure in

# VS Code which you may wish to be included in version control, so
this line

# is commented out by default.

#.vscode/

# Flutter/Dart/Pub related

**/doc/api/
**/ios/Flutter/.last_build_id
.dart_tool/
.flutter-plugins
.flutter-plugins-dependencies
.packages
.pub-cache/
.pub/
/build/

# Symbolication related

app.*.symbols

# Obfuscation related

app.*.map.json
```

```
# Android Studio will place build artifacts here  
/android/app/debug  
/android/app/profile  
/android/app/release  
  
# Miscellaneous  
*.class  
*.log  
*.pyc  
*.swp  
.DS_Store  
.atom/  
.buildlog/  
.history  
.svn/  
migrate_working_dir/  
  
# IntelliJ related  
*.iml  
*.ipr  
*.iws  
.idea/  
  
# The .vscode folder contains launch configuration and tasks you  
configure in
```

```
# VS Code which you may wish to be included in version control, so  
this line  
# is commented out by default.  
.vscode/  
  
# Flutter/Dart/Pub related  
**/doc/api/  
**/ios/Flutter/.last_build_id  
.dart_tool/  
.flutter-plugins  
.flutter-plugins-dependencies  
.packages  
.pub-cache/  
.pub/  
/build/  
  
# Symbolication related  
app.*.symbols  
  
# Obfuscation related  
app.*.map.json  
  
# Android Studio will place build artifacts here  
/android/app/debug  
/android/app/profile
```

```
/android/app/release  
name: qr_attendance_system  
description: A new Flutter project.  
  
# The following line prevents the package from being accidentally  
published to  
# pub.dev using `flutter pub publish`. This is preferred for private  
packages.  
publish_to: 'none' # Remove this line if you wish to publish to  
pub.dev  
  
# The following defines the version and build number for your  
application.  
# A version number is three numbers separated by dots, like 1.2.43  
# followed by an optional build number separated by a +.  
# Both the version and the builder number may be overridden in  
flutter  
# build by specifying --build-name and --build-number, respectively.  
# In Android, build-name is used as versionName while build-number  
used as versionCode.  
# Read more about Android versioning at  
https://developer.android.com/studio/publish/versioning  
# In iOS, build-name is used as CFBundleShortVersionString while  
build-number is used as CFBundleVersion.  
# Read more about iOS versioning at
```

```
#  
https://developer.apple.com/library/archive/documentation/General/R  
eference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html  
# In Windows, build-name is used as the major, minor, and patch  
parts  
# of the product and file versions while build-number is used as the  
build suffix.  
version: 1.0.0+1
```

environment:

```
sdk: '>=2.18.5 <3.0.0'
```

```
# Dependencies specify other packages that your package needs in  
order to work.
```

```
# To automatically upgrade your package dependencies to the latest  
versions
```

```
# consider running `flutter pub upgrade --major-versions`.
```

Alternatively,

```
# dependencies can be manually updated by changing the version  
numbers below to
```

```
# the latest version available on pub.dev. To see which dependencies  
have newer
```

```
# versions available, run `flutter pub outdated`.
```

dependencies:

```
flutter:
```

```
sdk: flutter

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.

cupertino_icons: ^1.0.2
firebase_core: ^2.4.1
firebase_auth: ^4.2.5
firebase_database: ^10.0.9
cloud_firestore: ^4.4.4
qr_flutter: ^4.0.0
flutter_barcode_scanner: ^2.0.0
local_auth: ^2.1.6
shared_preferences: 2.0.20
# qr_code_scanner:

dev_dependencies:
  flutter_test:
    sdk: flutter

# The "flutter_lints" package below contains a set of recommended
lints to
# encourage good coding practices. The lint set provided by the
package is
```

```
# activated in the `analysis_options.yaml` file located at the root of  
your
```

```
# package. See that file for information about deactivating specific  
lint
```

```
# rules and activating additional ones.
```

```
flutter_lints: ^2.0.0
```

```
# For information on the generic Dart part of this file, see the
```

```
# following page: https://dart.dev/tools/pub/pubspec
```

```
# The following section is specific to Flutter packages.
```

```
flutter:
```

```
# The following line ensures that the Material Icons font is
```

```
# included with your application, so that you can use the icons in
```

```
# the material Icons class.
```

```
uses-material-design: true
```

```
# To add assets to your application, add an assets section, like this:
```

```
assets:
```

```
  - assets/
```

```
    # - images/a_dot_ham.jpeg
```

```
# An image asset can refer to one or more resolution-specific  
"variants", see
```

```
# https://flutter.dev/assets-and-images/#resolution-aware
```

```
# For details regarding adding assets from package dependencies,  
see
```

```
# https://flutter.dev/assets-and-images/#from-packages
```

```
# To add custom fonts to your application, add a fonts section here,  
# in this "flutter" section. Each entry in this list should have a  
# "family" key with the font family name, and a "fonts" key with a  
# list giving the asset and other descriptors for the font. For  
# example:
```

```
# fonts:
```

```
#   - family: Schyler
```

```
#     fonts:
```

```
#       - asset: fonts/Schyler-Regular.ttf
```

```
#       - asset: fonts/Schyler-Italic.ttf
```

```
#         style: italic
```

```
#   - family: Trajan Pro
```

```
#     fonts:
```

```
#       - asset: fonts/TrajanPro.ttf
```

```
#       - asset: fonts/TrajanPro_Bold.ttf
```

```
#         weight: 700
```

```
#
```

```
# For details regarding fonts from package dependencies,
```

```
# see https://flutter.dev/custom-fonts/#from-packages
```

Key Screen Interfaces:

Admin's Screen:(figure 19)

Admin Dashboard Log Out



SEPM
lab



Physics

This is the Physics class for
Sem 1



Joined Users List



swetha

efrtNJXE0xXTnVWkda97NLfgLx
H3

+ Add New Class

← Add New Class ✓

◀ SEPM ▶

Class Name

Class Name (Mandatory)

Class Description

Class Description (Optional)



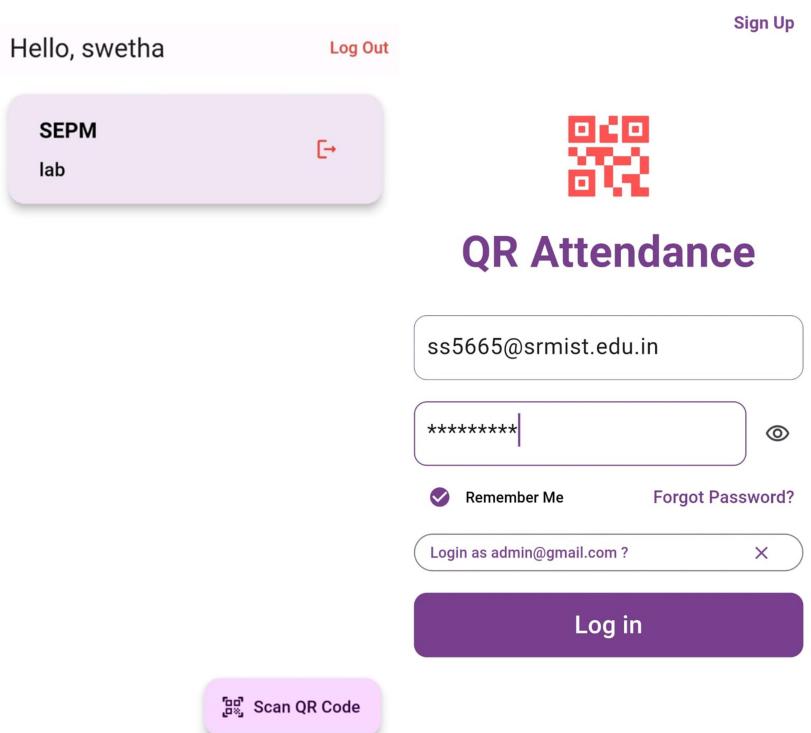
lab

Student List >

Submit

hlszTUQd2jKuwpMxRV1o

Student's screen:figure 20)



Result:

Thus, the details of architectural design/framework/implementation along with the screenshots were provided.

Conclusion: the proposed QR code-based attendance system project can revolutionize the way attendance is tracked in educational institutions and organizations. By automating the attendance tracking process, the system can save time, reduce errors, and improve data accuracy. The project's success will depend on its ability to address the unique needs of different institutions and organizations, making it a highly customizable and adaptable solution.

References:

- 1]"Design and Implementation of QR Code-Based Attendance Management System for Secondary Schools," by S.O. Popoola and O. Olaniyi: <https://www.sciencedirect.com/science/article/pii/S187704281404142X>
- 2]"QR Code-Based Attendance System for Higher Education Institutions," by S. Ravikumar, N. Jaya, and V. Thirumalai: <https://www.ijitee.org/wp-content/uploads/papers/v8i5/E9868058520.pdf>
- 3]"QR Code-Based Attendance System Using Android Application," by J. Soya and S. Saravanan: <https://ieeexplore.ieee.org/document/8452882>
- 4]"Implementation of a QR Code-Based Attendance Monitoring System in Tertiary Institutions," by F. Yusuf, I. A. Ganiyu, and O. O. Adejumo: https://www.researchgate.net/publication/331158114_Implementation_of_a_QR_Code-Based_Attendance_Monitoring_System_in_Tertiary_Institutions
- 5]"Development of a QR Code-Based Attendance System for the Classroom Environment," by J. R. Canete, F. R. Reyes, and D. L. Valencia: <https://ieeexplore.ieee.org/document/8321303>