Teachers Attendance System with fingerprint – Work Breakdown Structure (WBS)

Level 1: 1.0 Teachers Attendance System with fingerprint

Level 2: Major Phases

1.1 System Analysis & Planning
1.2 System Design
1.3 System Development
1.4 System Testing & Validation
1.5 Deployment & Documentation
1.6 Project Closure

Level 3: Tasks under Each Phase

1.1 System Analysis & Planning

- 1.1.1 Gather requirements
  - 1.1.1.1 A desktop application.
  - 1.1.1.2 Fingerprint-based authentication.
  - 1.1.1.3 Automated attendance reporting (daily, weekly, monthly, and term-based).
  - 1.1.1.4 Substitute teacher module.
  - 1.1.1.5 LAN-based deployment.
  - 1.1.1.6 Data encryption and secure storage for attendance and fingerprint information.

- 1.1.2 Identify hardware and software needs
  - 1.1.2.1 Fingerprint Scanners Devices.
  - 1.1.2.2 Desktop Application Platform.
  - 1.1.2.3 Database and Encryption Requirements.
  - 1.1.2.4 Existing LAN Compatibility.
  - 1.1.2.5 Cloud Backup .

- 1.1.3 Description database schema (SQLite)
  - 1.1.3.1 Serverless and File-Based :SQLite does not require a separate server process. The entire database is stored in a **single file (.db)**

  - 1.1.3.2 Ease of Deployment Since the database is a single file, deployment involves simply placing the file on a central school server accessible via the LAN.
  - 1.1.3.3 Security and Backup Simplicity.
  - 1.1.3.4 Excellent C# / .NET Compatibility:SQLite integrates seamlessly with the chosen **C#/.NET Desktop Application Platform**.
- 1.1.4 Create project schedule and assign tasks

1.2 System Design

1.2.1 System Architecture

- **1.2.1.1 The system uses a three-tier desktop architecture (UI – Logic – Database)**

- **1.2.1.2 All components run locally within the school's LAN for better security and performance**

- **1.2.1.3 The application supports multiple user roles: admin, management, and teacher**

1.2.2 Database Design

- **1.2.2.1 The database stores teacher profiles, attendance logs, and fingerprint data**

- **1.2.2.2 Data is managed through a relational model using the SQLite database system.**

- **1.2.2.3 Sensitive data such as fingerprint templates is encrypted for protection**

1.2.3 User Interface (UI)

- **1.2.3.1 The interface includes login, attendance recording, and reporting screens**

- **1.2.3.2 The design focuses on simplicity and ease of use for school staff**

- **1.2.3.3 Reports are accessible through a clear, organized admin dashboard**

1.2.4 Fingerprint Module

- **1.2.4.1 Integrates a fingerprint scanner using a compatible SDK**

- **1.2.4.2 Verifies teacher identity instantly and records attendance automatically**

- **1.2.4.3 Fingerprint data is stored locally in encrypted form**

1.2.5 Security & Testing

- **1.2.5.1 Implements AES encryption and role-based access control**

- **1.2.5.2 Conducts integration testing to ensure modules communicate correctly**

- **1.2.5.3 Verifies accuracy of attendance records before deployment**

1.3 System Development

- **1.3.1 Setting up the Development Environment**

Installing and configuring the core toolset: Visual Studio IDE and the .NET Framework for the C# Desktop Application ,Installing SQLite drivers, and integrating the Fingerprint Reader SDK(Software Development Kit) to enable communication between the software and hardware.

- **1.3.2 Database Creation and Configuration**
- **1.3.3 Developing the User Interface (UI)**
- **1.3.4 Implementing System Functions**

- **1.3.5 System Integration**

1.4 System Testing & Validation

- 1.4.1 Unit testing (individual functions)

- This testing is primarily performed by the developers themselves because they possess the deepest understanding of the code's internal structure and logic. This allows them to create exhaustive test cases .

- 1.4.2 Integration testing (Flutter + SQLite)

This testing ensures that the Flutter application(UI) can reliably and accurately read, write, update, and delete data from the SQLite database. It verifies the seamless flow of information from the UI layer to the data layer, confirming that essential application operations function without integration errors.

- 1.4.3 User acceptance testing (UAT)

  This testing performed by school IT Administration and teachers (users).
- 1.4.4 Bug fixing and optimization

QA Team:reports the bug and verifies that it is truly fixed after the developer finishes.

Development Team: fixes the code to eliminate the reported issue.

1.5 Deployment & Documentation

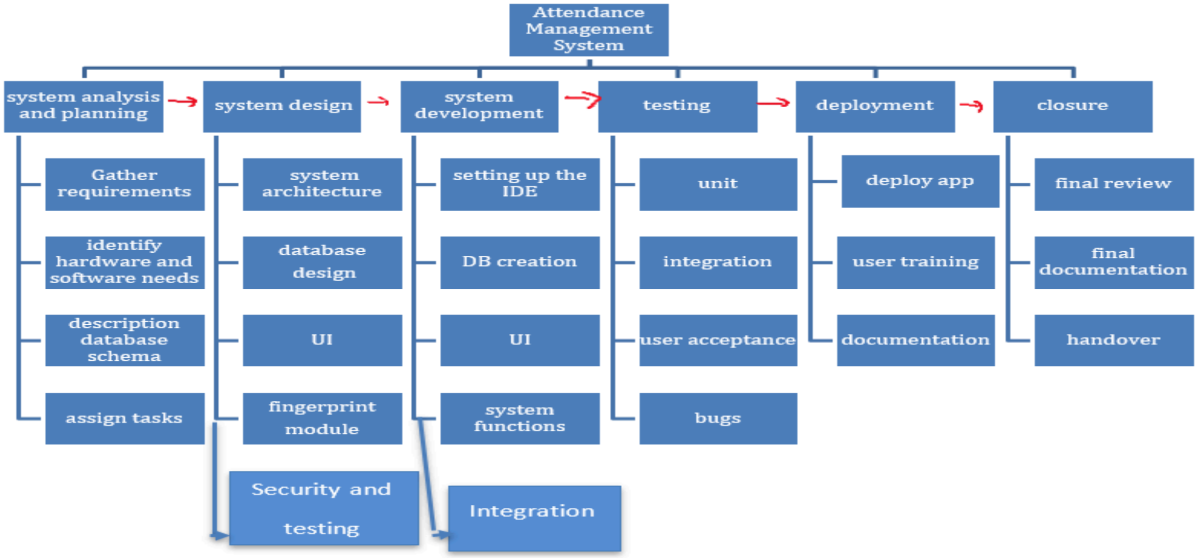
- 1.5.1 Deploy app

Install the desktop app on school computers.
Test to make sure the app works correctly.

- 1.5.2 User training
  Train school staff on how to use the app.
  Explain how to record attendance using fingerprints and check reports.
- 1.5.3 Documentation
  Create a simple user manual for staff.
  Prepare basic technical guide for developers.

1. 6 Project Closure

- 1.6.1 Final review meeting and approval by Project Sponsor and Project Manager,attended by IT Staff.
- 1.6.2 Submit final report and documentation
- 1.6.3 Handover to school IT Administration



WBS DICTIONARY

Code	Task Description	Assigned People	No. of Resources	Budget (LE)	Milestone (Date)
1.1.1	Gather requirements (first aid topics, content types)	Ahmed Nasser (Analyst), Sarah Fathy (Researcher)	2	1,0000	05/03/2025
1.1.2	Identify hardware and software needs	Omar Khaled (System Analyst)	1	5000	10/03/2025
1.1.3	Define database schema (SQLite)	Mariam Said (DB Designer)	1	8000	10/03/2025
1.1.4	Create a project schedule and assign tasks	Laila Mostafa (Project Manager)	1	7000	15/03/2025
1.2.1	System architecture	Hossam Magdy (UI/UX Designer), Jana Ali (Graphic Assistant)	2	1,200	25/03/2025
1.2.2	Design database tables and relationships	Mariam Said (DB Designer)	1	7000	25/03/2025
1.2.3	User interface	Jana Ali (Designer), Youssef Kamal (Frontend Dev)	2	1,000	5/04/2025
1.2.4	Fingerprint module	Laila Mostafa (PM), Ahmed Nasser (Analyst)	2	5000	5/04/2025
1.2.5	Security and testing	Rania Fathy (Tester), Ahmed Salah (QA Engineer)	2	10000	20/04/2025
1.3.1	Setting up the development environment	Youssef Kamal, Reem Tamer (Developers)	2	2,500	20/04/2025
1.3.2	Database creation and configuration	Mariam Said (DB Dev), Ali Hassan (Backend Dev)	2	2,000	25/04/2025
1.3.3	Developing the user interface	Reem Tamer, Jana Ali	2	1,000	5/05/2025
1.3.4	Implementing system functions	Ali Hassan (Backend Dev)	1	8000	5/05/2025
1.3.5	System integration	Ali Hassan (Backend Dev) Mariam Said (DB Dev) Rania Fathy (Tester)	3	15000	
1.4.1	Unit testing (individual functions)	Rania Fathy (Tester), Ahmed Salah (QA Engineer)	2	1,000	20/05/2025
1.4.2	Integration testing (Flutter + SQLite)	Ahmed Salah (QA), Youssef Kamal (Dev)	2	8000	30/05/2025
1.4.3	User acceptance testing	Ministry of Health team (3 reviewers)	3	7000	5/06/2025
1.4.4	Bug fixing and optimization	Developers team	2	1,000	15/06/2025
1.5.1	Deploy app	Ali Hassan (Dev), Laila Mostafa (PM)	2	1,200	25/06/2025
1.5.2	Conduct user training	Sarah Fathy (Trainer)	1	6000	5/07/2025
1.5.3	Prepare user manuals and technical documentation	Mariam Said (Tech Writer), Jana Ali (Designer)	2	1,000	10/07/2025
1.6.1	Final review meeting and approval	Laila Mostafa (PM), Ministry of Health Rep.	2	5000	15/07/2025
1.6.2	Submit final report and documentation	PM + Team	3	4000	20/07/2025
1.6.3	Handover to Ministry of school IT administration	PM + IT Admin	2	600	**25/7/2025**

Responsibility Matrix

WBS Code & Task	Laila (PM)	Ahmed (Analyst)	Sarah (Researcher)	Omar (System Analyst)	Mariam (DB Dev)	Hossam (UI/UX)	Jana (Designer)	Youssef (Frontend)	Ali (Backend)	Reem (Mobile)	Ahmed S. (QA)	Rania (Tester)	MOH Team
Gather requirements	S	R	R	S	–	–	–	–	–	–	–	–	–
Identify hardware & software needs	S	R	–	R	–	–	–	–	–	–	–	–	–
Define database schema (SQLite)	–	S	–	S	R	–	–	–	–	–	–	–	–
Create project schedule & assign tasks	R	S	–	–	–	–	–	–	–	–	–	–	–
Design UI/UX	S	–	–	–	–	R	S	–	–	–	–	–	–
Design database tables & relationships	–	–	–	S	R	–	–	–	–	–	–	–	–
Prepare prototypes	–	–	–	–	–	R	R	S	–	–	–	–	–
Review & approval of design	R	S	–	–	S	–	–	–	–	–	–	–	–
Frontend development (Flutter)	S	–	–	–	–	S	–	R	–	S	–	–	–
Backend development (SQLite)	–	–	–	–	R	–	–	–	R	–	–	–	–
Integrate text, images, and videos	–	–	–	–	S	S	R	R	–	S	–	–	–

Implement offline access feature	–	–	–	–	S	–	–	–	R	–	–	–	–
Unit testing (individual functions)	–	–	–	–	–	–	–	S	S	S	R	R	–
Integration testing (Flutter + SQLite)	–	–	–	–	S	–	–	S	S	–	R	R	–
User acceptance testing (MOH approval)	S	–	–	–	–	–	–	–	–	–	S	–	R
Bug fixing & optimization	–	–	–	–	S	–	–	R	R	S	S	–	–
Deploy app (Desktop)	R	–	–	–	S	–	–	S	R	S	–	–	–
Conduct user training	S	–	R	–	–	–	–	–	–	–	–	–	S
Prepare manuals & documentation	S	–	–	–	R	–	R	–	–	–	–	–	–
Final review & approval meeting	R	S	–	–	–	–	–	–	–	–	–	–	R
Submit final report & documentation	R	S	–	–	S	–	–	–	–	–	–	–	–
handover to school IT administration	R	–	–	–	–	–	–						