Teacher Helper

Abstract

Teacher helper is mobile application works on the android operating system. It helps teachers or instructors in managing attendance of students. It should help in recording attendance of students by fingerprint for each student. Then every student can view attendance sheet for the whole semester. A student can also help teacher by scanning the excuse of him/her if he/she is absent and send it to the teacher by the application. If the excuse is acceptable, it is automatically corrected from red to green. Teacher Helper project aims to manage the attendance of students, presenting absence causing and saving time and effort of teachers and students.

Abstract (in Arabic)

مسـاعد المعلم هو تطبيق يعمل على نظام التشـغيل أندرويد. يسـاعد المعلمين أو المدربين في إدارة حضـور الطلاب. يسـاعد في تسـجيل حضـور الطلاب من خلال بصـمات الأصـابع لكل طالب. يسـاعد كل طالب على عرض حضور الفصل الدراسي كامل. يساعد الطالب لعمل لرفع سبب الغياب وارساله للمعلم من خلال التطبيق. إذا كان عذر مقبول، يصحح تلقائيا من اللون الأحمر إلى اللون الأخضر. ويهدف مشروعنا إلى إدارة حضور الطلاب ، تقديم أعذار الغياب ، وتوفير الوقت والجهد للمعلمين والطلاب.

Keywords

Mobile application, Teacher Helper

List of Abbreviations

TH: Teacher Helper.

SDLC: Software Development Life Cycle.

ERD: Entity Relationship Diagram.

DFD: Data Flow Diagram

GUI: Graphical User Interface

JKD: Java developer kit.

ADT: Android Development Tool.

SDK: Software Development Kit.

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

Introduction

This chapter discusses the problems, scope, purpose, objectives, business requirement, Alternative Solutions and System Perspective of teacher helper application.

Teacher helper application is a mobile application used in education that helps the teacher or instructor in managing the attendance of the students.

1.1: Project Overview

Teacher helper is a mobile application. It shall be used in the education field. it helps teacher or instructor to manage student's attendance. It should help to take the attendance of the students by fingerprint in class of each student. Then it is shown to the student the attendance sheet of all the semester as a green block if she attends or red block if she absents. The application can also help the committee excuse by upload the excuse of the student when she absents and sends it to the committee by the application. If the excuse is accepted, it automatically changes from red to green. This application does not cover student grade participation (track record) and submission.

1.2: Problem Statement

Every class, the teacher has to take attendance manually by calling students names or by taking their signature of in attendance sheet. This procedure takes a long time from the lecture and effort of the in structure. In the case of absent, the student needs to look for the teacher to give her the absence excuse. sometimes the teacher is absence or doesn't have time to meet the student.

If continue to use this procedure, they will not only be wasting time and effort. Also, the student will be prohibited from the exam, when excused absence is lost.

1.3: Project Impact

Implementing an android application system instead of the traditional manual system is to save time and effort of teachers and students in recording attendance. This system will help to manage the attendance efficiently and easily through the use of the mobile application. It also facilitates communication between commission excuse, teachers, and students.

1.4: Project Scope

The teacher has ZKT fingerprint device connect to his computer, when student attendance lecture, he will record fingerprint by ZKT device. The application update status of the student to the attendant.

The application enables the student to send absence excuse. The employee view absence excuse if he accepts it, the attendance sheet will update from red to green.

Student details exist in faculty system. When a teacher selects any course to record attendance, the application will view all students in this course. So, we don't need to add student details.

1.5: Objectives

- Teacher helper helps teacher managing absence record of students.
- It facilitates communication between excuse committee, students, and teacher.
- The application will help to manage attendees of the student.
- Save time and effort with teachers and students.

1.6: Business Requirements

The application easy to the teacher and student for access to information accurately and at high speed and Communicate with the committee excuses effective and easy way.

- The application saves excuses from loss, also helps the student to know the number of absences and the number of hours remaining for each subject.
- The application process is to take the attendance to be more safety.
- The application helps the student to attach excuse and save it technically
- The application easily instructor on the process of taking attendance technically
- The application supports the link between teacher and student and the Committee Excuses.
- The application allows the student to take more than fingerprint If you cannot use one of the fingerprints due to injuries or other.
- The application allows the student using fingerprint.
- The application supports fingerprint conformity and verification.

Another solution

Different users have different expectations of the system behavior, and all these combined together constitute the overall system requirements

Stakeholder of the application:

- Teachers
- Students
- Employee (Commit excuse employee)
- Admin

Admin

- The admin shall be able to log in by user id and password.
- The admin shall be able to assign courses for each teacher.
- The admin shall be able to add fingerprint for each student.
- The admin shall able to view a detail student.
- The admin shall able to delete a detail student.

Employee

- The employee shall be able to log in by user id and password.
- The employee shall be able to check absence cause.
- The employee shall be able to view attendance sheet.

Teacher

- The teacher shall be able to log in by user id and password.
- The teacher shall be able to view attendance sheet.
- The student shall able to record attendance.

Student

- The student shall able to log in by user id and password.
- The student shall able to record attendance.
- The student shall able to add absence cause.
- The student shall able to view attendance sheet.

1.7: Alternative Solutions

Android provides freely its Software Development Kit (SDK) to the developer community which minimizes the development and licensing costs. Get the open source

advantage from licensing, royalty-free, and the best technology framework offered by the Android community. The entire platform is ready for customization. You can integrate and tweak the mobile app according to your business need. Android applications can be deployed in different ways. Android applications are scripted in Java language with the help of a rich set of libraries. So, Teacher Helper project is an android application

There is alternatives solution of this android application such as

- Used IOS application.
- Connect the application with blackboard system.
- Barcode reader device

Another solution is costly. We did not use IOS application for this reasons

- Not flexible only supports iOS devices
- Not open source
- The main disadvantages of using iOS are costly Apps and no widget support
- Devices are very pricey
- Applications are very large when compared to other mobile platforms

1.8: System Perspective

Limitation to success of Teacher Helper project

- Use android devices only.
- Minimum android 2.3 is needed.
- Failed in any student device.
- Provide ZKT device for each teacher.
- Facilitated the spread of viruses on the network

Factors to success Teacher Helper application

- Use more than one ZKT device.
- Use antivirus.

Summary

This chapter discusses the problems, scope, purpose, objectives, business requirement, Alternative Solutions and System Perspective of teacher helper application. In next chapter, we will introduce literature review of another system.

Chapter two: Background

Background

This chapter discusses the literature review, Existing Business Processes, Proposed Business Process, Methodology, and Project Planning.

A literature review can be referring to as a review of the current system that the researcher had done previously and the review of the system that will be developed. The literature review also focuses on the knowledge and ideas established on a topic as well as their Advantages and weaknesses. Nowadays, technology is getting better and better to replacing the traditional system to speed up the process by introducing the computerized system. There are few types of attendance system that had been introduced nowadays in school, college, and university.

2.1 Overview of existing system

We search in articles about biometrics technologies that use in recording attendance. Biometrics technologies verify identity through characteristics such as fingerprints, faces, irises, retinal patterns, palm prints, voice, hand-written signatures, and so on. These techniques, which use physical data, are receiving attention as a personal authentication method that is more convenient than conventional methods such as a password or ID cards. The biometric personal authentication uses data taken from measurements. Such data is unique to the individual and remains so throughout one's life.

It is important to identify the correct tools to use in record attendance. Barcode readers, Radio Frequency Identification (RFID) system, Bluetooth, and NFC are just a few of the examples of such tools. However, they were expensive when first introduced and therefore they had limited use.

We read many of the previous articles that talk about how to record attendance of student to get an idea about how to implement Teacher Helper project. This article focuses on the knowledge and ideas established on a topic as well as their Advantages and weaknesses. There are some of this article:

2.1.1 Kadry S. et al., "Wireless Attendance Management System based on Iris Recognition ", 2010.

System iris recognition is one of the ways to make the attendance of students/staffs to be manageable in more effective and automated ways. In the iris recognition, the system will require acquiring the iris image of students/staffs and store inside the database after going through several processes by the system.

There are some Advantages can be found in the attendance system that implements the iris-recognition feature.

- Iris recognition is irreplaceable and it is unique for everyone.
- The iris is placed behind the cornea and eyelid which make it always protected from the external environments, not like other biometric methods.
- It will definitely help improve the staff morale, increase the work efficiency, and create a good business image of the company. On the other hand, for students, it will help in reduce the poor attendance of the student and make them attend all the classes obediently.
- It uses wireless technique rather than the wired-based which can help in reducing the risk of malfunctions of the background wired-based machine as the iris-recognition will be performed wirelessly.

However, there is some drawback that can be found in the solution in implement the wireless iris recognition attendance system.

- The devices for the iris recognition is quite expensive compared to any other biometric recognition devices.
- Implementation of wireless technique will require high transmission speed of network in order to verify the students/staffs identify.

2.1.2 Reda A. et al., “A Low-cost Remote Attendance Tracking System for Developing Regions”, 2011.

The system is a combination of voice recognition technology and location tagging in obtaining the attendance of the remote staffs. The system will be developed in the mobile platform in order to reduce the hardware cost of obtaining the attendance of the remote agents. In addition, location tagging tools will be implemented to the system as

well to track the location of the remote agents while they make verification for their attendance.

There is some strength that can be found in the attendance system that using voice recognition technology.

- The system was developed in the mobile platform in order to reduce the deployment costs.
- Voice recognition is known as the less privacy-sensitive forms of verification compared to other biometric forms such as fingerprint, palm print, and etc. So, remote agents do not need to worry so much about the loss of privacy issue in sending their voice for verification.

There is some drawback that can be found from the solution of using voice recognition technology in the attendance system.

- The system will require the remote agents to have a smartphone in order to verify their attendance since the system was developed on a mobile platform.
- The cellular network must be available in using this system since it will take the attendance remotely and Global Positioning System (GPS) will be required in order to track the location of the users while they verify their attendance.
- It is still possible for voice recognition technology to occur errors in verification as voice can be easily affected by the body condition.

2.1.3 Dhanashree A. G., “Student Attendance Management”, 2011.

The system only requires basic equipment such as a set of desktop computer. In short, the proposed system only requires the users to install the software to their laptop/desktop for managing their student attendance. There are two modules introduced which are admin module for managing the classes and report module for generating the attendance report.

There is some strength that can be found in the desktop-based attendance system.

- The system can be known as a computerized system that reduces the workload of the users in managing the attendance records.

- It will also help the lecturers save up a lot of time in recording the attendance of the students throughout the whole semester.

- It does not require any special hardware to make it workable, so it will be very easy to use due to its simplicity.

However, there is some drawback that can find the solution in using desktop-based student attendance system.

- The system had only converted the traditional attendance system to computerization but everything still has to be done manually such as key-in student attendance and etc.

- The system does not save up much time from the record the student attendance.

2.1.4 Microtronics Technologies," RFID Based Attendance Management System", 2013.

The RFID system is developed and is suitable to take the attendance of the students as well as employees. There are two modules introduced in the article which includes reader module and RFID module. In details, each student/employee must have a valid RFID card of RFID tags with them in order to communicate with the RFID reader placed on their workplace/school. The RFID reader will automatically detect the student/employee attendance and record it while the RFID card gets closer to the RFID reader which means it is using the non-contact type of reader and passive types of card.

There is some strength that can be found in the attendance system that using RFID technology.

- The system can be known as a fully-automated system which requires less human interaction in the attendance record process.

- The user will be able to view the attendance lists on the spot through the computer interface or view remotely through another computer.

However, there is some drawback that can find the solution of using RFID technology in the attendance system.

- The system will require the student/employee to bring the RFID cards always with them while they are in class or workplace in order to check-in or check-out for the attendance. If the student/employee lost their card, they may need to

go to the office to make a new card which will cause them to pay for the lost as well as waiting for the new card to be generated and pass to them.

- However, although the system can help prevent the student from making fake attendance but nobody can guarantee that the student will not take their friend's RFID cards along with them to the classes which mean it still gets the same result which scanning the cards on behalf of their buddies.

2.1.5 Subramaniam H. et al., "Bar Code Scanner Based Student Attendance System (SAS)", 2013.

In the barcode scanner technology, the student will be issued a student card for each of them with the barcode displayed on the card for a scanning purpose every time they attend the classes. Student attendance status will be automatically checked and record into the system once lecturer scans their student card with a barcode scanner.

There is some strength that can be found in the attendance system that using barcode scanner technology.

- The system provides a report module which allows the lecturer to generate a daily, weekly and monthly report.
- The system also provides a functionality which is if the student does not meet the attendance requirements, the system will automatically generate warning letter to the student to be delivered to their parent.
- Barcode scanner requires less cost development compare to the RFID technology.

However, there is some drawback that can find the solution in using barcode scanner technology in the attendance system.

- The system will require the student to have their student card with the unique barcode displayed on their student card every time they attend a class. So sometimes if the student forgot to bring their student card along with them while going to the class which may cause the lecturer will need to go to the office to ask the admin staff to change their student attendance status. It is because lecturer is not authorized to manually key-in the student attendance as the

system will automatically update the student attendance status into the database after scan the student card.

- Although the system allows the admin staff to generate warning letter if the student does not meet the attendance requirements, however, the generated warning letter will be given to the student and then pass it to their parent through the student themselves without giving an instant message or email to their parent. So, the student may end up with just throw away the letter and pretending that they already surrender the warning letter to their parents.

2.1.6 Mohammad A. et al., "Integrated System for Monitoring and Recognizing Students during Class Session", 2013.

The system will take a picture of the whole class by classroom's camera and upload to the system to do face filtering and then the attendance of the student will be checked automatically by the system once the face matching of a student is successfully performed. The system also allows the lecturer to drag and drop their student's picture into the system if the system failed to recognize their student's face.

There is some strength that can be found in the attendance system that using face recognition technology.

- The system provides a high-security feature since the face recognition will only start to record attendance by performing face filtering after the lecturer has a login using their ID and password and capture the picture of whole class.

- Sometimes the student face may not be recognized by the face detection system but the system provides a good feature which let lecturer upload the student photo that used to match the photo of the student stored in the database. By having this features, a student who does not recognize by the face detection would also be able to take the attendance.

- The system will reduce the percentage of fake attendance or maybe almost zero chances for student to make a fake attendance since everyone has a different look as their unique identity.

However, there is some drawback that can find the solution in using face recognition technology in the attendance system.

- The system will require the lecturer to manually upload a photo to the system in order to complete the attendance through face recognition process.
- Although it has provided the report features, but the report only can produce immediately after the attendance has done. The user is not able to print again if the report is lost. Although the record is backed up to the database server, but there are no web services provided for the user to print the report and it cause the user have to manually retrieve the data from the database server if they want to print the report again.
- The hardware for the face recognition is too expensive to implement at every class as it makes the system become invaluable anymore.

2.1.7 Talaviya G. et al., "Wireless Fingerprint Based College Attendance System Using Zigbee Technology ", 2013.

In the fingerprint recognition technology, students will require registering their fingerprint into the database for future matching while they attend every class.

There is some strength that can be found in the attendance system that using fingerprint recognition technology.

- The system had provided the report generation module which will automatically send to few parties such as Head of Department (HOD), lecturers, and student's parent regarding their attendance status in the specified days.
- Due to the uniqueness of the fingerprint, buddy signing will be able to eliminate since they will be unable to mark the attendance on behalf of their friends.

However, there is some drawback that can find the solution in using fingerprint technology in the attendance system.

- The system did not provide a manual key-in feature in their attendance system which means those students who come late to the class with reason will still consider absent.

2.1.8 Comparisons between existing solution and proposed solution

After a literature review, we compare between this existing system and Teacher Helper project. This Comparison shall be made on these factors:

- Use unique identifier.
- Existing report generation.
- Existing email notification.
- Save time.
- Accurate and efficient attendance records.
- Student Lateness Detection.
- Eliminate Buddy-Signing.
- Scan excuse absent.

We found the most existing system use the unique identifier to record attendance, record attendance in this system is accurate, and this system eliminates buddy-signing. Some of this system is used email notification and report generation. All this system does not have a method to send excuse absent.

Comparisons	Unique identifier	Report Generation	Email Notification	Save Time	Accurate and efficient attendance records	Student Lateness Detection	Eliminate Buddy-Signing	Scan excuse absent
Wireless Attendance Management System based on Iris Recognition	YES	YES	NO	YES	YES	NO	Yes	NO
A Low-cost Remote Attendance Tracking System for Developing Regions	YES	NO	NO	NO	NO	NO	YES	NO
Student Attendance Management	NO	NO	NO	NO	NO	NO	NO	NO

RFID Based Attendance Management System	NO	NO	YES	YES	YES	NO	NO	NO
Bar Code Scanner Based Student Attendance System (SAS)	NO	NO	NO	YES	YES	NO	NO	NO
Integrated System for Monitoring and Recognizing Students during Class Session	YES	YES	NO	NO	NO	NO	YES	NO
Wireless Fingerprint Based College Attendance System Using Zigbee Technology	YES	YES	NO	YES	YES	NO	YES	NO
Teacher helper *(Proposed System)*	YES	YES	No	YES	YES	YES	YES	YES

Table 2.1 Comparisons between existing solution and proposed solution

2.2 Existing Business Processes

In Imam University teacher takes attendance manually by calling students names or by the signature of a student in attendance sheet this wasted a long time from the lecture. The student search about a teacher to present absence excuse, the sometimes teacher is absence or teacher doesn't have time to meet the student.

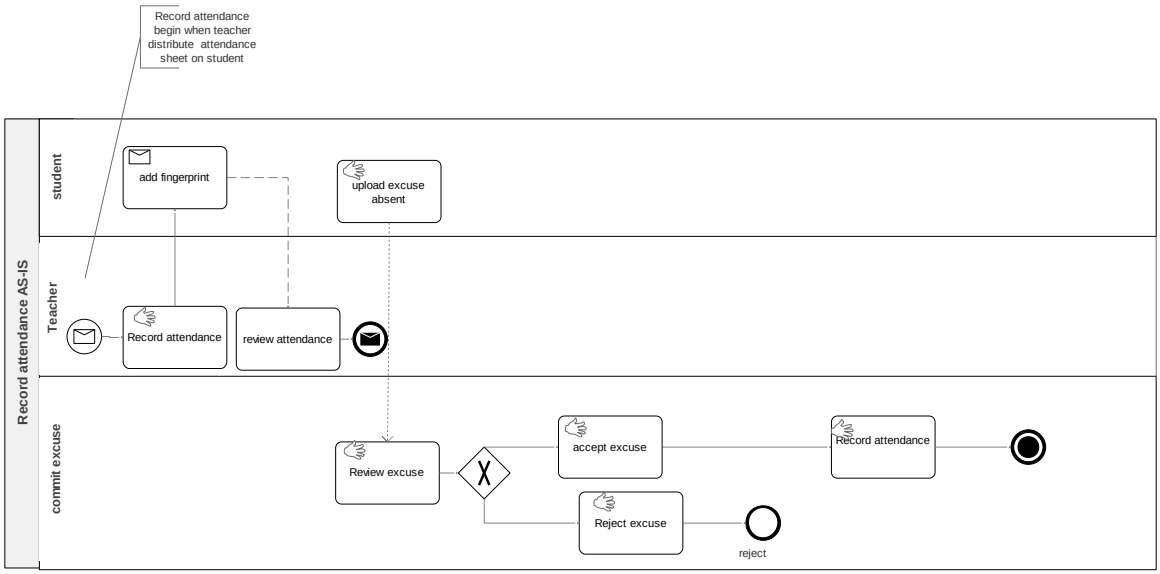


Figure 2.1 The record attendance As-IS

2.3: Method / Approach [1]

The SDLC has a similar set of four fundamental phases: planning, analysis, design, and implementation.

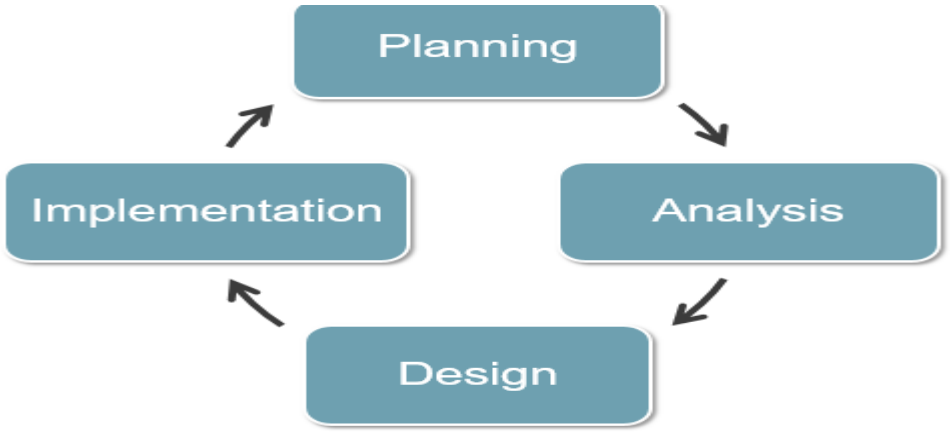


FIGURE 2-2 Systems Development Life Cycle

Planning

The planning phase is the fundamental process of understanding why an information

The system should be built and determine how the project team will go about building it.

Analysis

The analysis phase answers the questions of who will use the system, what the system will do,

and where and when it will be used. During this phase, the project team investigates any current
system(s), identifies opportunities for improvement, and develops a concept for the new system.

Design

The design phase decides how the system will operate, in terms of the hardware, software, and network infrastructure; the user interface, forms and reports; and the specific programs, databases, and files that will be needed. Although most of the strategic decisions about the system were made in the development of the system concept during the analysis phase, the steps in the design phase determine exactly how the system will operate.

Implementation

The final phase in the SDLC is the implementation phase, during which the system is actually
built. This is the phase that usually gets the most attention because for most systems it is the longest and most expensive single part of the development process.

Waterfall Model

The Waterfall model is a conventional, linear, sequential or traditional waterfall software life cycle model. It is a sequential development approach, in which development is seen as flowing steadily downwards through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

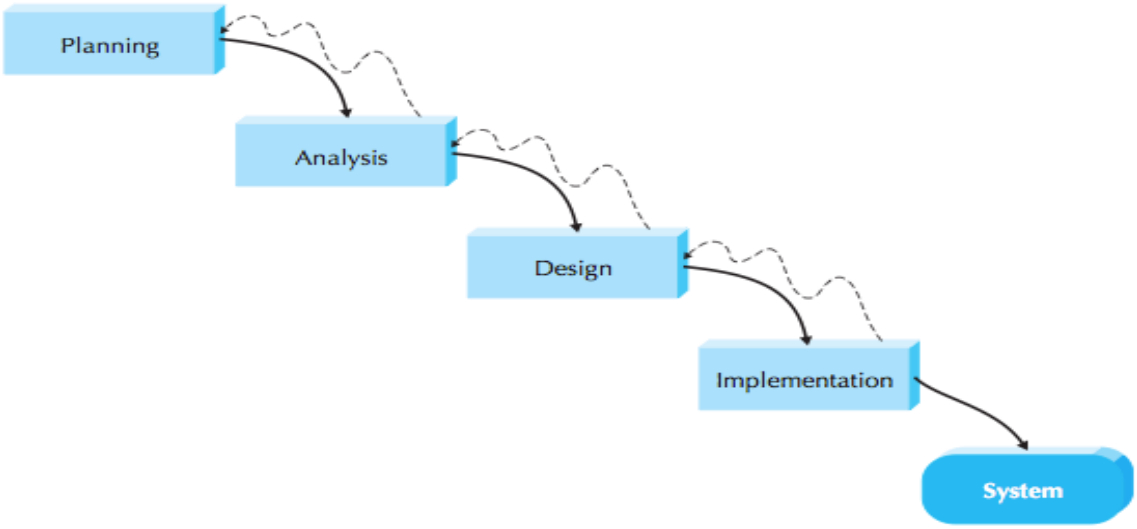


Figure 2.3 waterfall model

Advantages of waterfall model

- Simple and easy to use it.
- It is suitable for small projects.
- We can call it the base of the models.

Disadvantages of waterfall model

- The worst model for large projects.
- The Biggest disadvantage is not meet the new customer requirements.
- Difficult to correct mistakes on each step.

We opted to use the waterfall life cycle for a number of reasons:

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.

- Process and results are well documented.

2.4 Project Plan

Project plan describes phases of Teacher Helper project and the time determined for each phase.

Teacher Helper project includes four phases planning, analysis, design, and implementation phase. Each phase includes some titles this title shown in (figure 2.4).

2.4.1 WBS chart

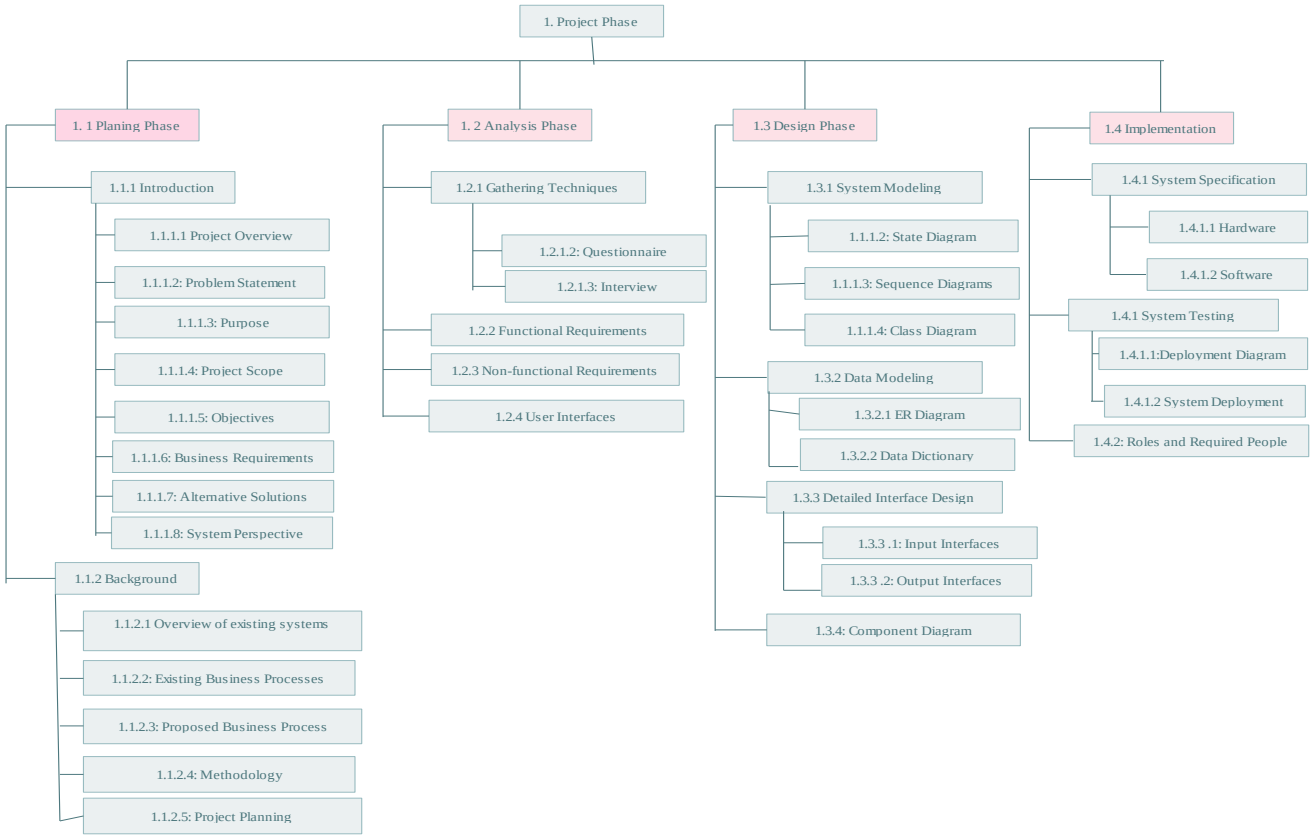


FIGURE 2-4 WBS chart

2.4.2 Identify the project tasks.

In this table explain responsibility for each member of our group, start date, end date, and status of this responsibility. All members participate in each responsibility in the project. Some of this responsibility is completed, and the other will complete in next semester.

Responsibility	Start date	Finish date	Team member	Status
Project overview	10/1/2016	10/1/2016	All team members	Completed
Problem statement	10/2/2016	10/2/2016	All team members	Completed
Project scope	10/3/2016	10/3/2016	All team members	Completed
Project objective	10/4/2016	10/5/2016	All team members	Completed
Literature review	10/6/2016	10/6/2016	All team members	Completed
methodology	10/7/2016	10/7/2016	All team members	Completed
Project plan	10/8/2016	10/9/2016	All team members	Completed
Estimate cost	10/10/2016	10/11/2016	All team members	Incomplete
Requirements	10/12/2016	10/16/2016	All team members	Incomplete
Questionnaire	10/17/2016	10/20/2016	All team members	Incomplete
Gather information	10/21/2016	10/24/2016	All team members	Incomplete
Use case diagram	10/25/2016	10/30/2016	All team members	Incomplete
Activity diagram	11/1/2016	11/5/2016	All team members	Incomplete

Design diagrams	11/6/2016	11/10/2016	All team members	Incomplete
Interface design	11/11/2016	11/12/2016	All team members	Incomplete
Hardware and software specification	11/13/2016	12/20/2016	All team members	Incomplete
Write codes	12/21/2016	3/23/2017	All team members	Incomplete
testing	3/24/2017	3/30/2017	All team members	Incomplete
Future work	4/1/2017	4/2/2017	All team members	Incomplete
Conclusion	4/2/2017	4/2/2017	All team members	Incomplete

TABLE 2-2 Members Responsibility

2.4.3 Gantt chart

Task Name	start date	end date	duration
Teacher Helper	**10/1/2016**	**5/5/2017**	**217**
Initiating	10/1/2016	10/5/2016	5
meeting	10/1/2016	10/1/2016	1
create a team member	10/2/2016	10/2/2016	1
define problem	10/3/2016	10/3/2016	1
proposal form	10/4/2016	10/5/2016	2
proposal	10/6/2016	10/16/2016	11
Meeting Assigning task to member	10/6/2016	10/6/2016	1
problem statements	10/7/2016	10/7/2016	1
scope, objectives	10/8/2016	10/9/2016	2
Hw/SW requirements	10/10/2016	10/11/2016	2
Report of proposal	10/12/2016	10/16/2016	5
Background	10/17/2016	10/20/2016	4

Methodology	10/21/2016	10/24/2016	4
System Analysis	10/25/2016	11/12/2016	19
information gathering	10/25/2016	10/30/2016	6
requirements definition	11/1/2016	11/5/2016	5
Function modeling	11/6/2016	11/10/2016	5
data modeling	11/11/2016	11/12/2016	2
System Design	11/13/2016	1/1/2017	50
Database design	11/13/2016	12/20/2016	38
Interface Design	12/21/2016	12/30/2016	10
Deployment Diagram	12/31/2016	12/31/2016	1
Alternative Design	1/1/2017	1/1/2017	1
System Implementation and Test	1/2/2017	4/1/2017	90
Discussion and conclusion	4/2/2017	4/20/2017	19
Finishing	4/21/2017	5/5/2017	15
Revision design	4/21/2017	5/1/2017	11
Final document	5/2/2017	5/5/2017	4

TABLE 2-3 Identify the duration for each task

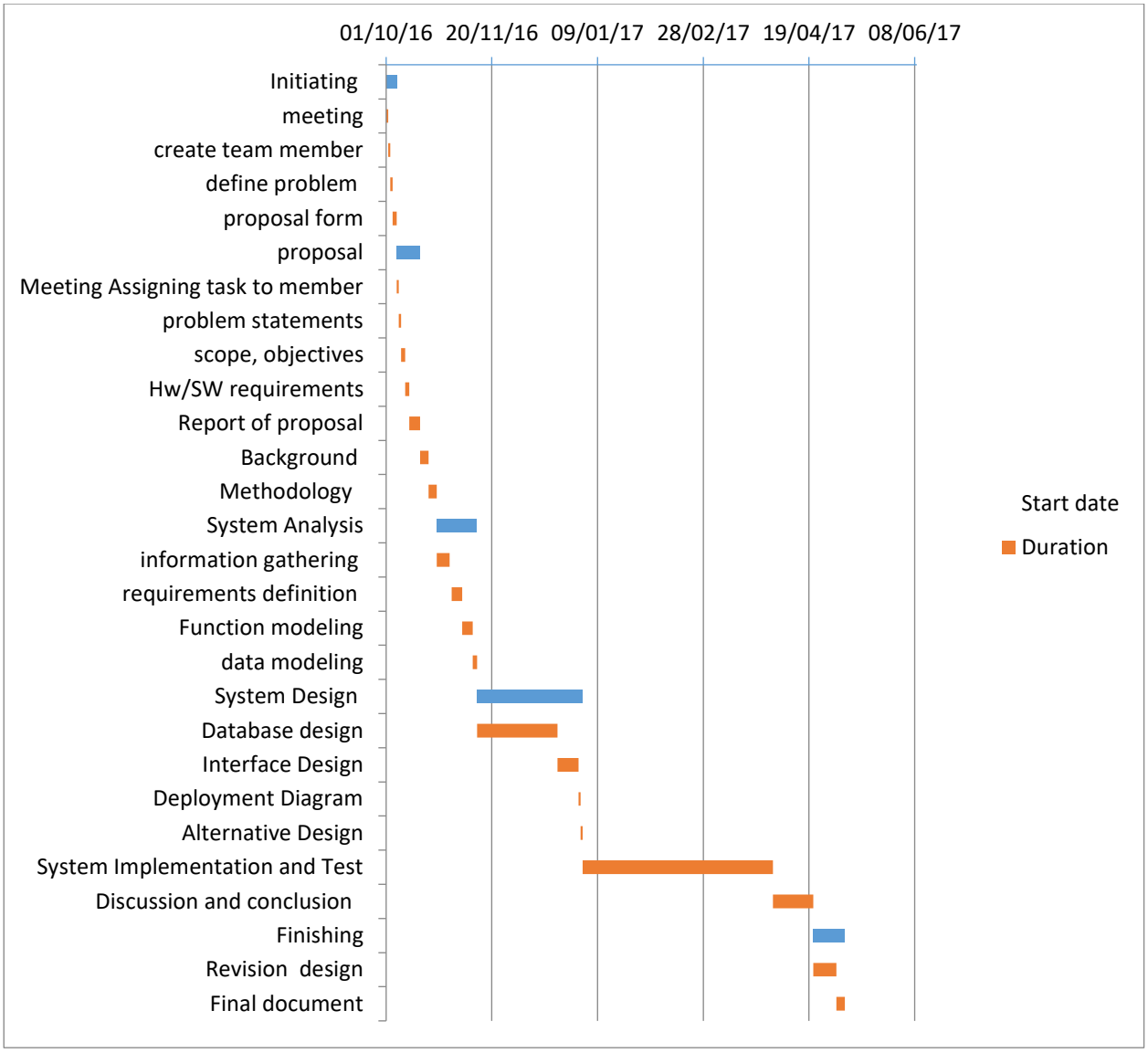


FIGURE 2-5 Gantt chart

conclusion

After gone through the literature review of the few attendance systems, several limitations or weaknesses of the previous researcher's solution had been identified. In the proposed system, the limitations and weaknesses found from the literature review will be improved and enhances in order to make the student attendance system operate more efficiently and effectively in assisting the lecturer to take attendance of the student.

Chapter Three: Requirements Analysis

Introduction

This chapter will cover the requirements analysis phase of Teacher Helper project that includes the techniques that we used to gather the information for developing Teacher Helper application then cover the requirements specifications (the functional and non-functional requirements), then mock interface for the proposed system

3.1: Requirement Gathering Techniques

The important step for building knowledge for a new system or to develop an existing system is information gathering. There are many techniques for information gathering but we prefer the Questionnaires are a good technique to gather requirements from remote locations. Questionnaires are an appropriate method to gather input from a huge number of people.

Results of Questionnaire

Information gathered from the questionnaire:

1. Most teachers need a system to record attendance.
2. Provide a report about attendance all semester.
3. Send an absence cause to employee if student absences.

3.2: Proposed Business Process

Record attendance begins when teacher login and send a broadcast signal about the occurrence of attendance registry, the student record attendance by fingerprint. The application enables the student to send absence excuse. The employee view absence excuse if he accepts it, he will send to the teacher and he will update attends sheet from red to green.

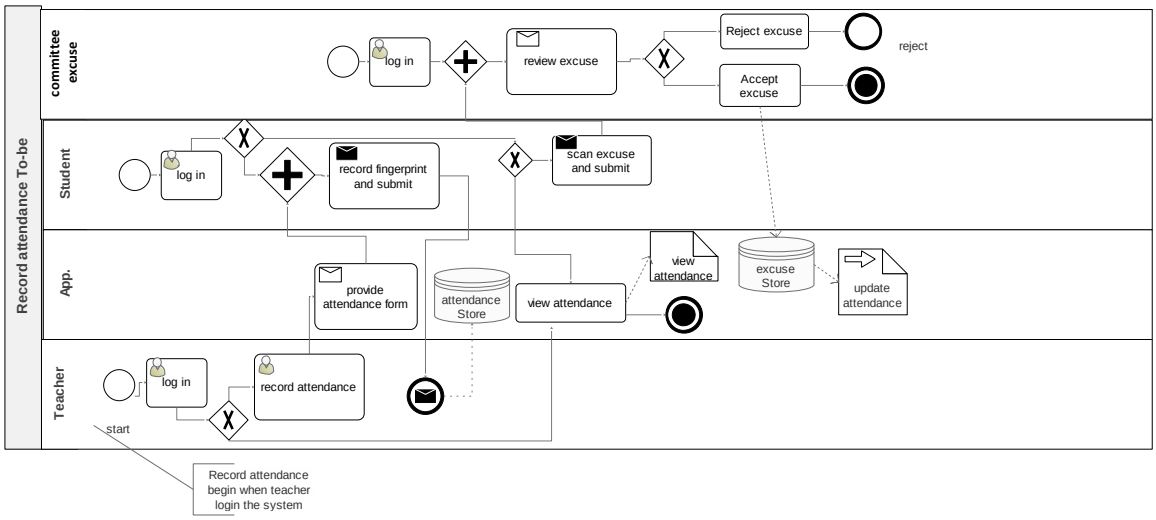


Figure 3.1The record attendance To-be

The architecture of Teacher Helper system. Hardware of Teacher Helper system consists of android devices of teacher and students such as mobile, tab. etc. Wi-Fi modem, and server to the database.

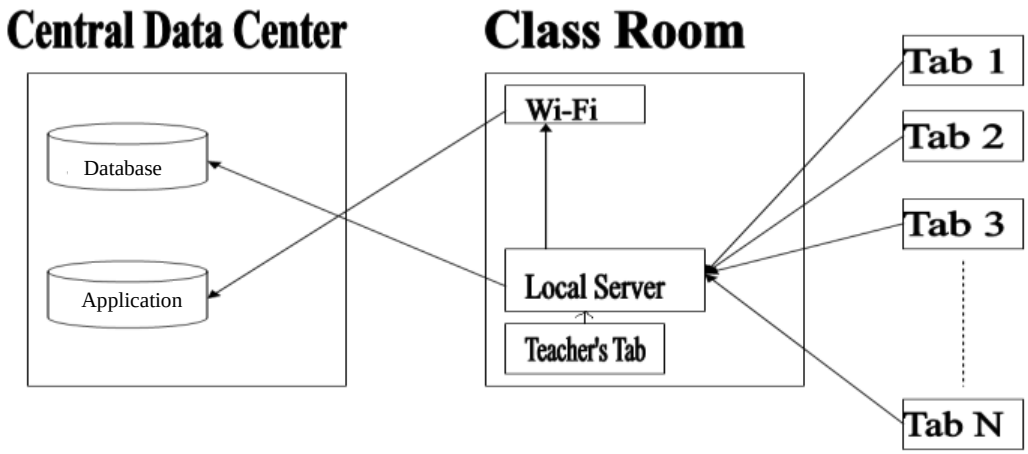


Figure 3.2 The architecture of teacher helper

When each student record fingerprint the system matches this fingerprint with these that had stored in database and show if it matching or not such as in (figure 3.3) [9].

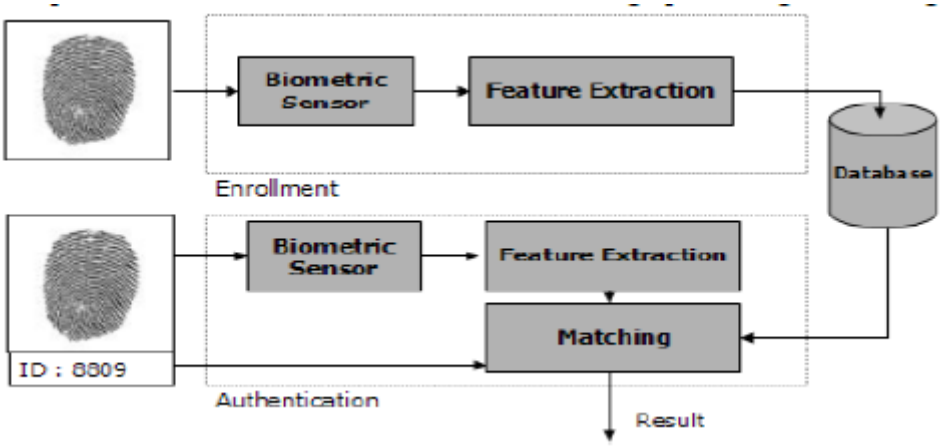


Figure 3.3 Block Diagram of Biometric Authentication

Flowchart of Teacher Helper system. The system begins record attendance when teacher login and activate record attendance link.

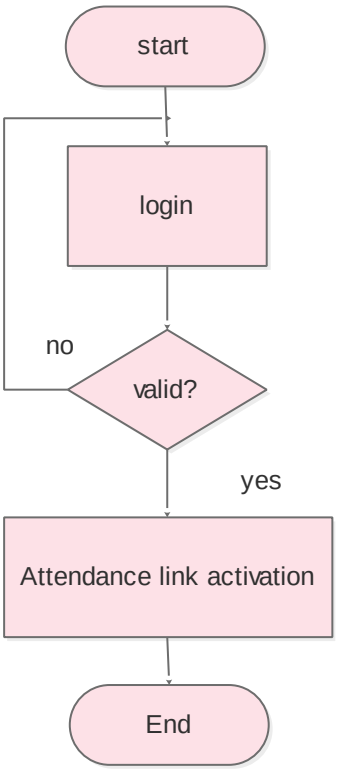


Figure 3.4 Flowchart for the teacher

The students log in the system and record fingerprint and the system will check this fingerprint.

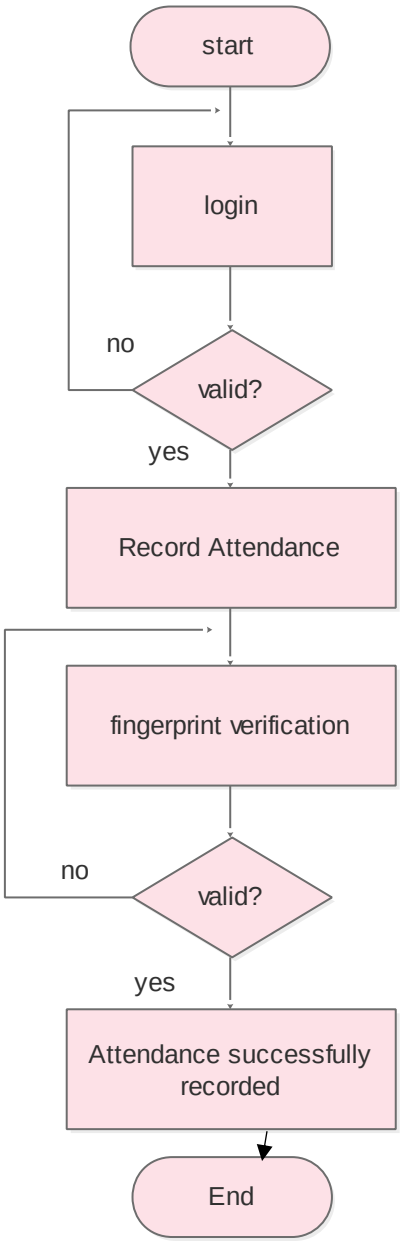


Figure 3.5 Flowchart for the student

3.3: Functional Requirements

1. **The admin, teacher, employee, and student should be able to login to the system.**

1.1. The system should provide an account for the teacher.

1.2. The system should provide an account for the admin.

1.3. The system should provide an account for the employee.

1.4. The system should provide an account for the student.

1.5. The system should contain a login form.

2. **The admin shall be able to assign courses for each teacher.**

2.1. The system should allow the admin to add assign a course to the teacher by using a form.

2.2. The system should add the assigned course to the database.

2.3. The system should provide a search engine.

2.4. The system should retrieve the search result from the database.

3. **The admin should manage the account for each student.**

3.1 The system should allow the admin to add fingerprint for each student using a form.

3.2 The system should add the new fingerprint to the database.

3.3 The system should provide a search engine.

3.4 The system should retrieve the search result from the database.

3.5 The system should allow the admin to view a student.

3.6 The system should retrieve student information from the database.

3.7 The system should allow the admin to delete a student.

4. **The employee shall be able to check absence cause.**

4.1 The system should allow the employee to view absence cause message using a form.

4.2 The system should allow the employee to accept absence cause using a form.

4.3 The system should add accept absence cause to the database.

4.4 The system should update attendance sheet.

4.5 The system should provide a search engine.

4.6 The system should retrieve the search result from the database.

4.7 The system should allow the employee to view attendance sheet.

4.8 The system should retrieve attendance sheet information from the database.

4.9 The system should retrieve student attendance from the database.

5. **The student shall able to record attendance.**

5.1 The system should allow the student to add student fingurprint using ZKT device.

5.2 The system should add student attendance to the database.

6. **The student shall able to add absence cause.**

6.1 The system should allow the student to attach absence cause using form.

6.2 The system should add absence cause to database.

7. **The student shall able to view attendance sheet.**

7.1 The system should allow student view attendance sheet using form.

7.2 The system should retrieve attendance sheet from database

8. **The teacher shall able to view attendance sheet.**

8.1 The system should allow teacher view attendance sheet using form.

8.2 The system should retrieve attendance sheet from database

Use-case diagram

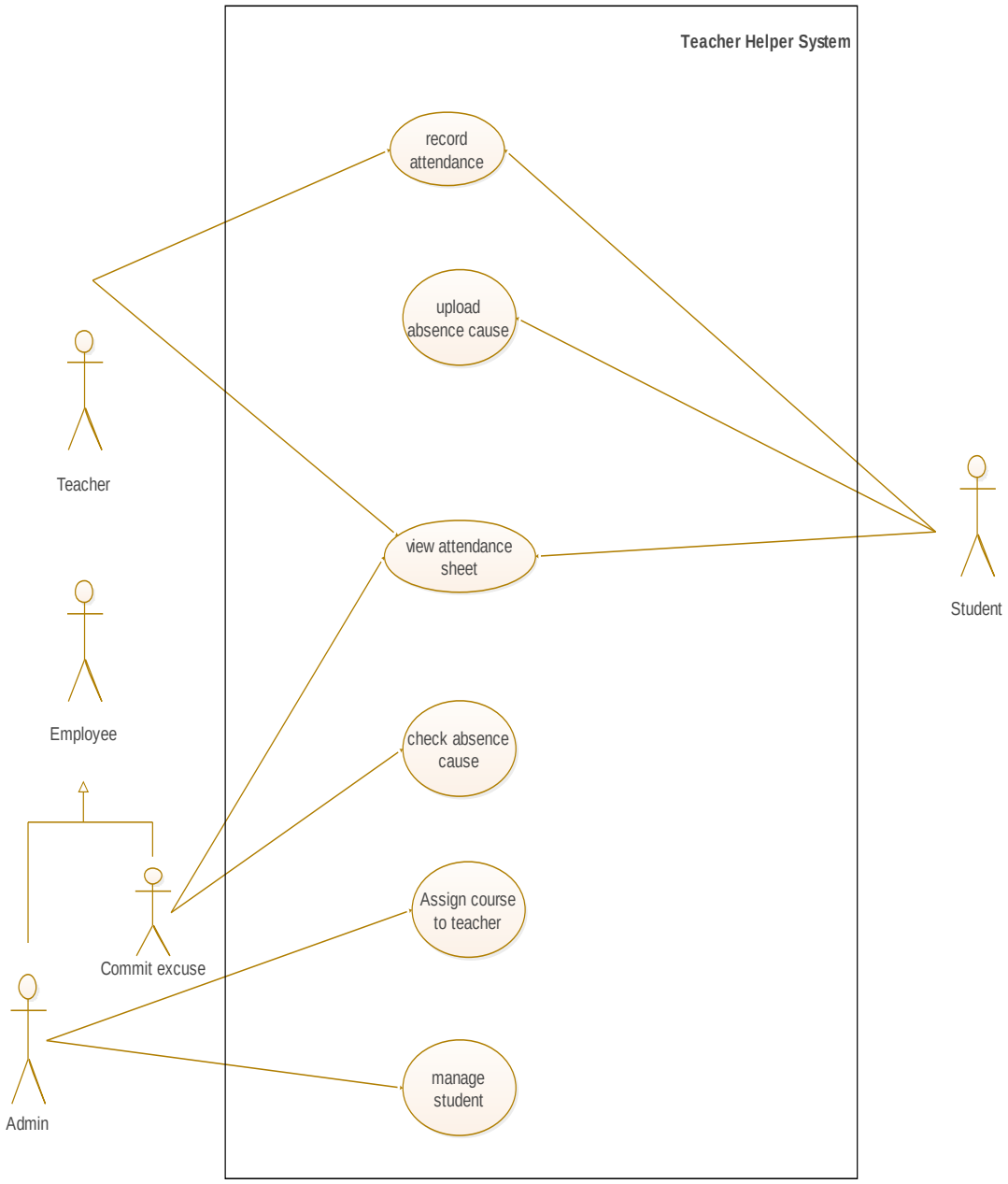


FIGURE 3.6 Use case diagram

Use case description for log in

Use case name: login	ID: 1	Important level: high
Primary actor: Employee, Teacher, Student, Admin		Use case type: Detail, essential
Stockholder and interests: The admin, employee, teacher, student want to log in the system.		
Brief description: this case describes how to log in the system.		
Triggers: user clicks login button. Type: external		
Relationship: Associative: Employee, Teacher, Student, Admin Included: Extended: yes Generalization:		
The normal flow of event: 1. The user enter username and password. 2. The system checks validate username and password. 3. The system opens the main page to a user according to his/her privilege.		
Sub flows:		
Alternative flow: 3. the system asks user to re-enter username and password if not correct. 4. Go to step1		

TABLE 3.1 Use case description for log in

Use case description for assign course to teacher

Use case name: assign course to teacher	ID: 2	Important level: high
Primary actor: admin		Use case type: Detail, essential
Stockholder and interests: Admin want to assign each course to his/ her teacher. Teacher: find all his /her courses in his /her account.		
Brief description: this case describes how to assign a course to the teacher.		
Triggers: admin click assign course to teacher button. Type: external		
Relationship: Associative: Admin Included: Extended: Generalization:		
The normal flow of event: 1. The admin login the system. 2. The system displays the main page. 3. The admin clicks on assign course to teacher button. 4. The system displays assign course form. 5. The admin selects course name, teacher name, and submit it. 6. The system check data. 7. The system saves data in the database.		
Sub flows:		
Alternative flow: 7. The system asks user to re-select course name, and teacher name if this teacher does not teach this course. 8. Go to step5		

TABLE 3.2 Use case description for assign course to teacher

Use case description for manage student.

<table>
<tr><td>Use case name: manage student.</td><td>ID: 3</td><td>Important level: high</td></tr>
<tr><td colspan="2">Primary actor: admin</td><td>Use case type: Detail, essential</td></tr>
<tr><td colspan="3">Stockholder and interests:

Employee: save student fingerprint in the database, view or delete student.

student: record attendance by fingerprint.</td></tr>
<tr><td colspan="3">Brief description: this case describes how to add fingerprint, view student, delete student.</td></tr>
<tr><td colspan="3">Triggers: admin clicks manage button.

Type: external</td></tr>
<tr><td colspan="3">Relationship:

Associative: admin

Included:

Extended:

Generalization:</td></tr>
<tr><td colspan="3">The normal flow of event:

1. The admin login the system.
2. The system displays the main page.
3. The admin clicks on manage button.
4. The system displays manage form.
If an admin, click add fingerprint.
The 4-1 add fingerprint sub flow is performed.
If an admin, click view student.
The 4-2 view student sub flow is performed.
If an admin, click delete student.
The 4-3 delete student sub flow is performed.
5. The system saves data in the database.</td></tr>
<tr><td colspan="3">Sub flows:</td></tr>
</table>

4-1 add fingerprint 1. The admin selects student ID, and ask the student to put his/her finger in right place. 2. The system checks data. 4-2 view student. 1. The admin enters student ID. 2. The system checks data and views student details. 4-3 delete student 1. The admin enters student ID. 2. The system checks data and views student details. 3. The admin clicks the delete button. 4. The system shows a confirmation message. 5. The system delete student from the database if employee clicks Yes.
Alternative flow: 4-1 2. The system ask student to re-enter fingerprint again 4-2 2. The system ask admin to re-enter student ID 4-3 2. The system ask admin to re-enter student ID

TABLE 3.3 Use case description for manage student.

Use case description for check absence cause.

Use case name: check absence cause	ID: 4	Important level: high
Primary actor: Employee		Use case type: Detail, essential
Stockholder and interests: Employee: accepting or unaccepting absence cause. student: update his/ her attendance sheet.		
Brief description: this case describes how to check absence.		
Triggers: employee click absence cause button. Type: external		

Relationship: Associative: Employee Included: Extended: Generalization:
The normal flow of event: 1. The employee login the system. 2. The system displays the main page. 3. The employee clicks on absence cause button. 4. The system displays absence cause form. 5. The employee view absence cause and click on accept button. 6. The system updates attendance sheet data in the database.
Sub flows:
Alternative flow: 5.the employee clicks on reject cause.

TABLE 3.4 Use case description for absence cause

Use case description for view attendance sheet.

Use case name: view attendance	ID: 5	Important level: high
Primary actor: Employee, student, teacher		Use case type: Detail, essential
Stockholder and interests: The employee, Teacher: view percentage of attendance for each student in the course. Student: view number of attendance and number of absent.		
Brief description: this case describes how to view attendance sheet.		
Triggers: user click view attendance button. Type: external		

Relationship: Associative: Employee, student, teacher Included: Extended: Generalization:
The normal flow of event: 1. The user login the system. 2. The system displays the main page. 3. The user clicks on view attendance button. 4. The system displays view attendance form. 5. The user fill form by selects course name, and student id. 6. The system views attendance sheet from the database.
Sub flows:
Alternative flow:

TABLE 3.5 Use case description for view attendance sheet

Use case description for take attendance.

Use case name: take attendance	ID: 6	Important level: high
Primary actor: teacher, student		Use case type: Detail, essential
Stockholder and interests: Teacher: activate record attendance button. Student: enable them to record attendance.		
Brief description: this case describes how to take attendance.		
Triggers: user turn on ZKT device. Type: external		
Relationship: Associative: teacher, students		

Included: Extended: Generalization:
The normal flow of event: 1. The teacher login the system. 2. The system displays the main page. 3. The teacher selects course name and date. 4. The student put finger on fingerprint screen. 5. The system check data. 6. The system saves data.
Sub flows:
Alternative flow: 12.the system asks student to re-enter fingerprint.

TABLE 3.6 Use case description for take attendance

Use case description for upload absence cause.

Use case name: upload absence cause	ID: 7	Important level: high
Primary actor: student		Use case type: Detail, essential
Stockholder and interests: Student: update record attendance.		
Brief description: this case describes how to add absence cause.		
Triggers: student click adds absence cause button. Type: external		
Relationship: Associative: student		

Included: Extended: Generalization:
Normal flow of event: 1. The student login the system. 2. The system displays the main page. 3. The student clicks on add absence cause button. 4. The system displays add absence cause form. 5. The student browses the directory which files saved in it and select it. 6. The student sends button. 7. The system saves data.
Sub flows:
Alternative flow:

TABLE 3.7 Use case description for upload absence.

3.3 NON-FUNCTIONAL REQUIREMENTS

Descriptions of the system's functions, services and operational constraints in teacher helper system.

1. Security

1.1 Registered users shall have a Login ID and Password.

1.2 The user shall be able to view and modify all information in the system.

1.3 The user information will never be sold to other parties and will be kept secure at all times.

1.4 Users will be authenticated to ensure that no unauthorized users gain access to private information.

2. Usability

2.1 The Graphic User Interface (GUI) of the system must be user-friendly.

2.2 The GUI of the system will present conceptual integrity.

2.3 The system will be developed so that it is easy to use a system that requires the least amount of user input possible.

2.4 Every input will be validated.

2.5 Error messages will be displayed if the user enters an invalid data or tries to access a function without the required permissions.

3. Supportability

3.1 The system should be supported by Google Groom, Mozilla Firefox, Internet Explorer and Safari browsers.

4. Implementation

4.1 Language

4.1.1 The system should be written in android studio language.

4.2 Platform

4.2.1 The system should operate on windows 7 onwards.

5. Predictability

5.1 The system should produce predictable results.

6. Reliability

6.1 The system should be capable of processing a 100 user within the same time frame with no errors and the system should be available and operational all the time.

7. Correctness

7.1 The system will be considered correct when the prototype presented and agrees that all the functions they require are implemented as stated in the Software Requirements Specification.

8. Flexibility

8.1 The system should be developed in such a way that it is easily customizable. If new functions are required, there will be the little effort required to update the system.

9.Availability

- The system shall meet or exceed 99.99% uptime.
- The system shall not be unavailable more than 1 hour per 1000 hours of operation.
- Less than 20 seconds shall be needed to restart the system after a failure 95% of the time.

10. Robustness

- The estimated loss of data in case of a disk crash shall be less than 0.01%.

- The system shall be able to handle up to 10000 concurrent users when satisfying all their requirements and up to 25000 concurrent users with browsing capabilities.

3.4: User Interfaces

FIGURE 3.7 home page

FIGURE 3.8 employee page

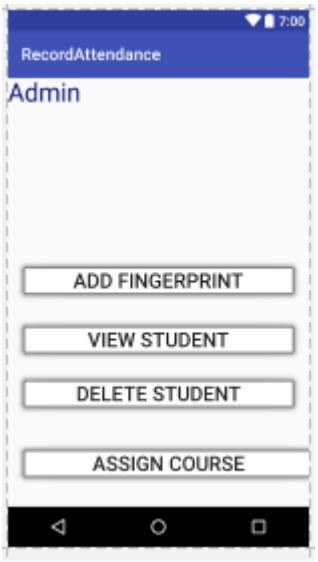


FIGURE 3.9 Admin page

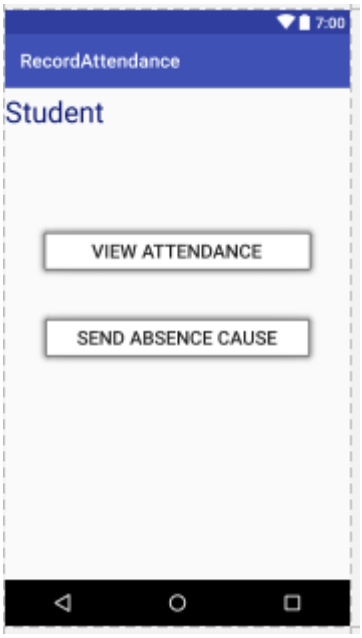


FIGURE 3.10 student page

FIGURE 3.11 employee course assignment page

FIGURE 3.12 Add fingerprint

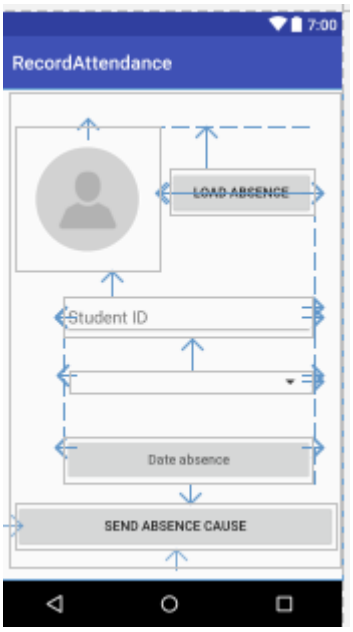


FIGURE 3.13 Send absence cause

FIGURE 3.14 View absence cause

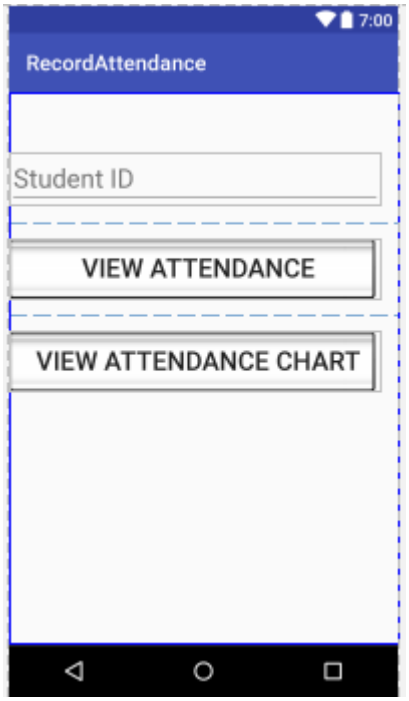


FIGURE 3.15 Teacher view attendance

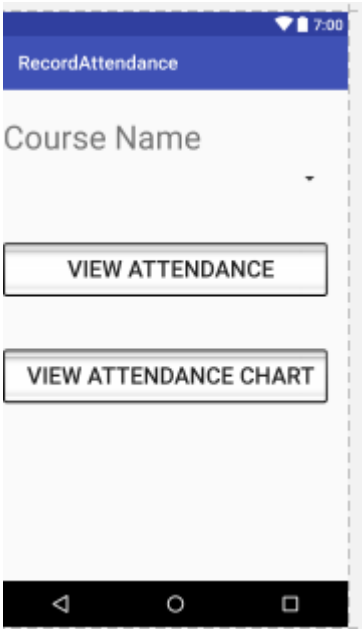


FIGURE 3.16 Student View attendance

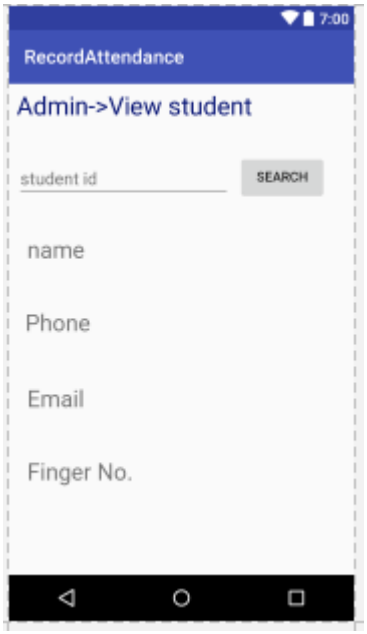


FIGURE 3.17 view student information

FIGURE 3.18 Delete student information

Chapter four: System Design

This chapter discusses the system modelling, Data Modelling, Detailed Interface Design, Component Diagram, Alternative design of teacher helper application.

Teacher helper application is a mobile application used in education that helps the teacher or instructor in managing the attendance of the students.

4.1: System Modelling

System modeling has now come to mean representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). System modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers

4.1.1: Activity diagrams for each use-case

An activity diagram visually presents a series of actions or flow of control in a system to describe the steps in a use case diagram

4.1.1.1 Assign course activity diagram

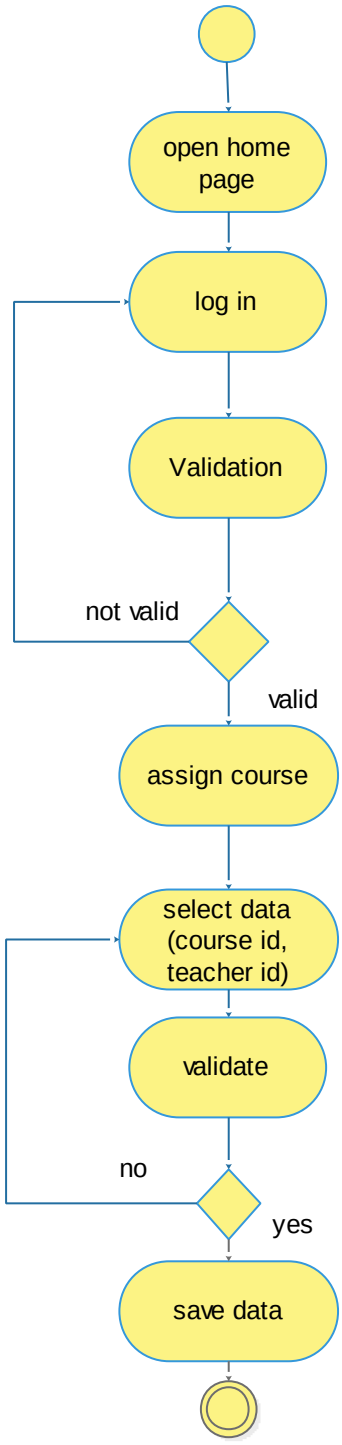


FIGURE 4. 1: Assign course activity diagram

4.1.1.2 Check absence activity diagram

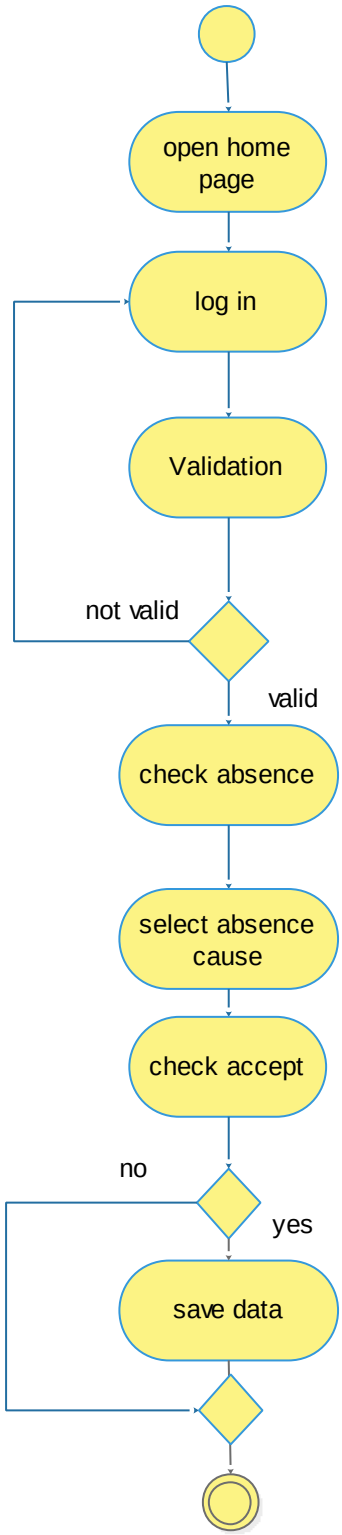


FIGURE 4. 2: Check absence activity diagram

4.1.1.3 View attendance sheet activity diagram

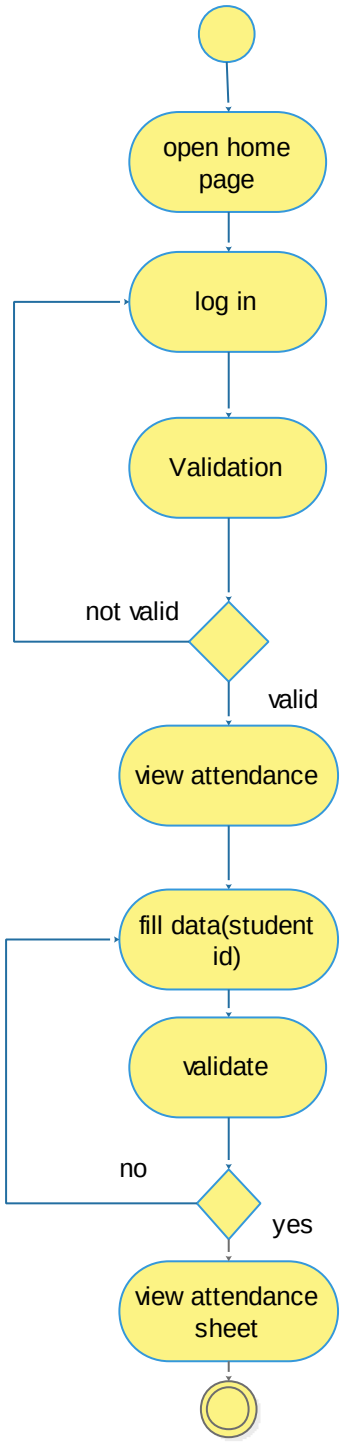


FIGURE 4. 3: View attendance sheet activity diagram

4.1.1.4 Upload absence activity diagram

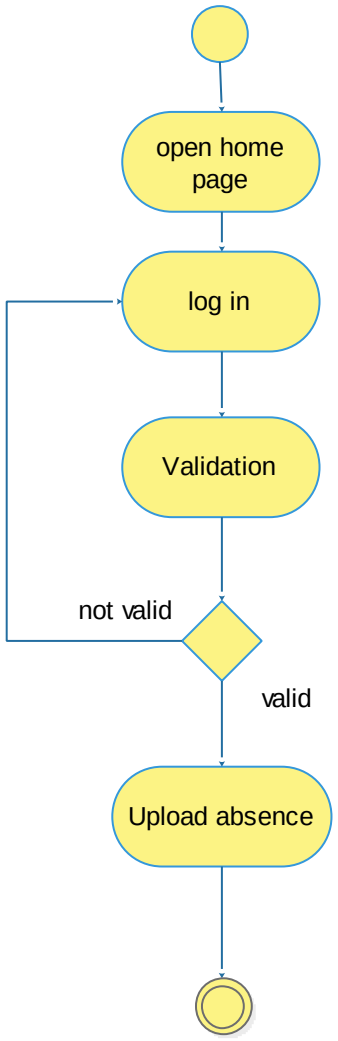


FIGURE 4. 4: Upload absence activity diagram

4.1.1.5 Manage student activity diagram

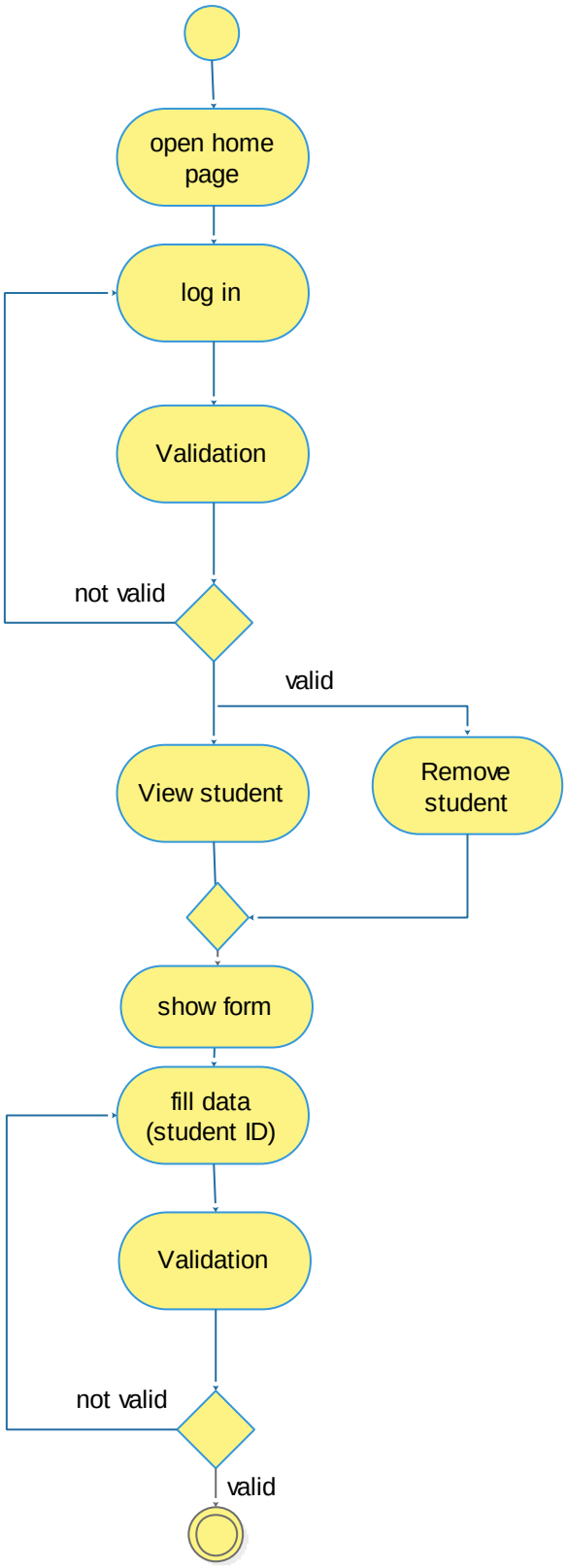


FIGURE 4. 5: Manage student activity diagram

4.1.2: Class Diagram

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

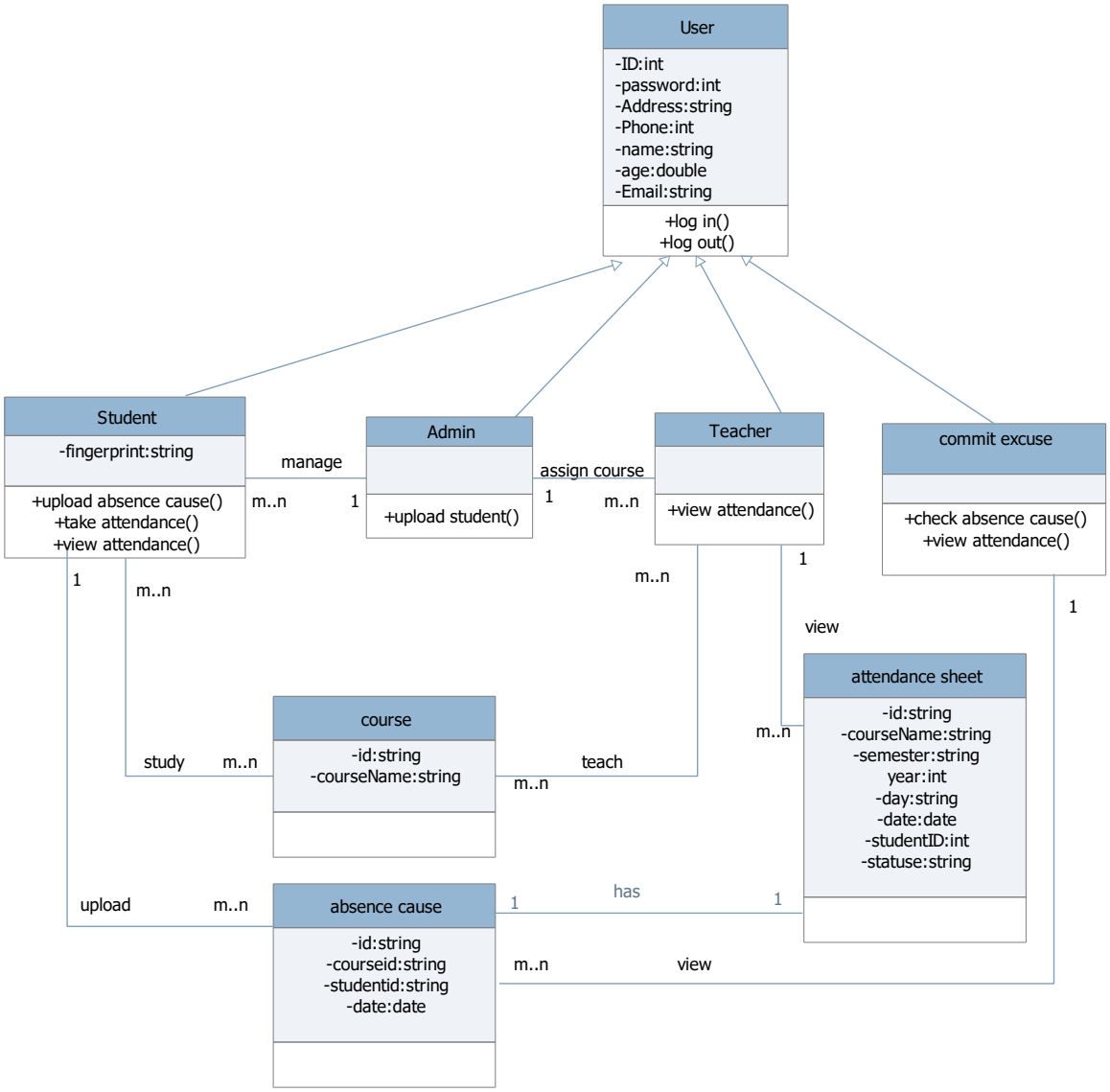


FIGURE 4. 6: class diagram

4.1.3: Sequence Diagrams

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. In this part we draw important sequence diagrams in teacher helper system such as add fingerprint, add attendance check absence cause, view attendance, and delete attendance.

4.1.3.1: Add fingerprint sequence diagram

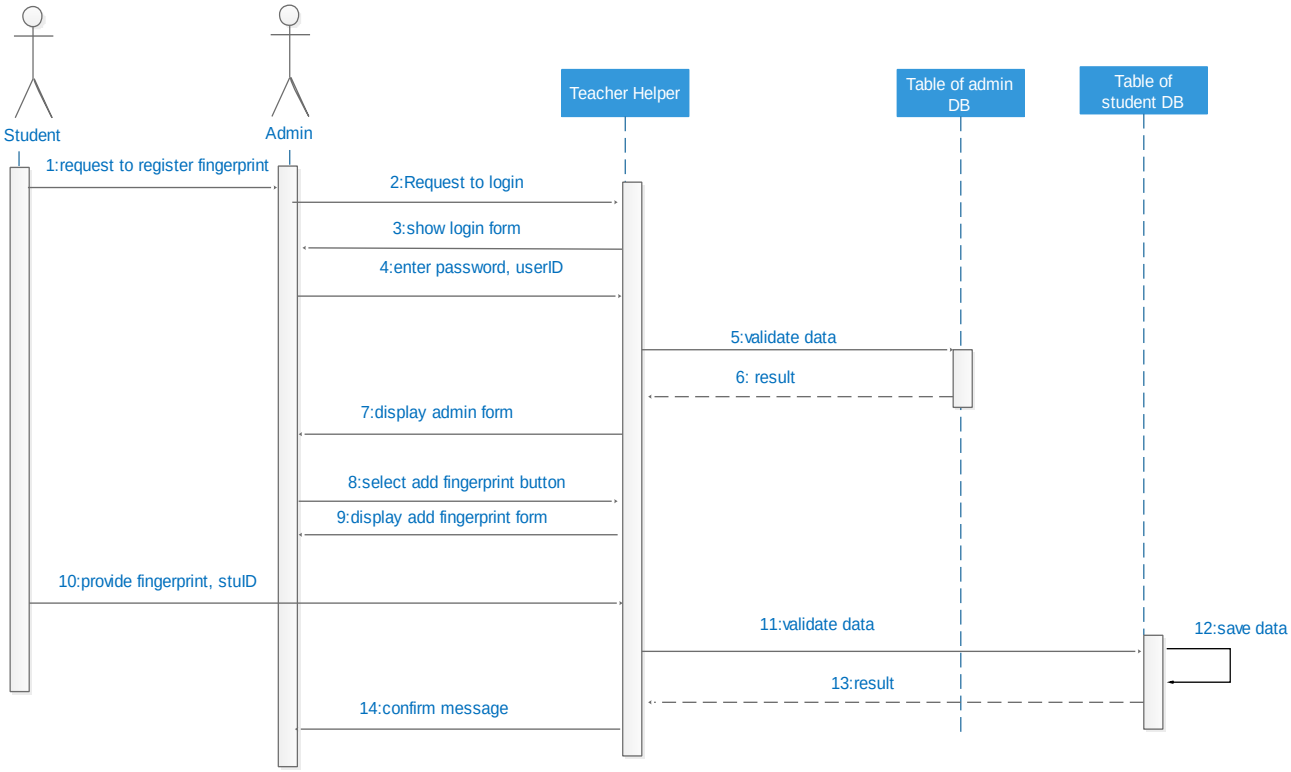


FIGURE 4. 7: Add fingerprint sequence diagram

4.1.3.2 Assign course to teacher sequence diagram

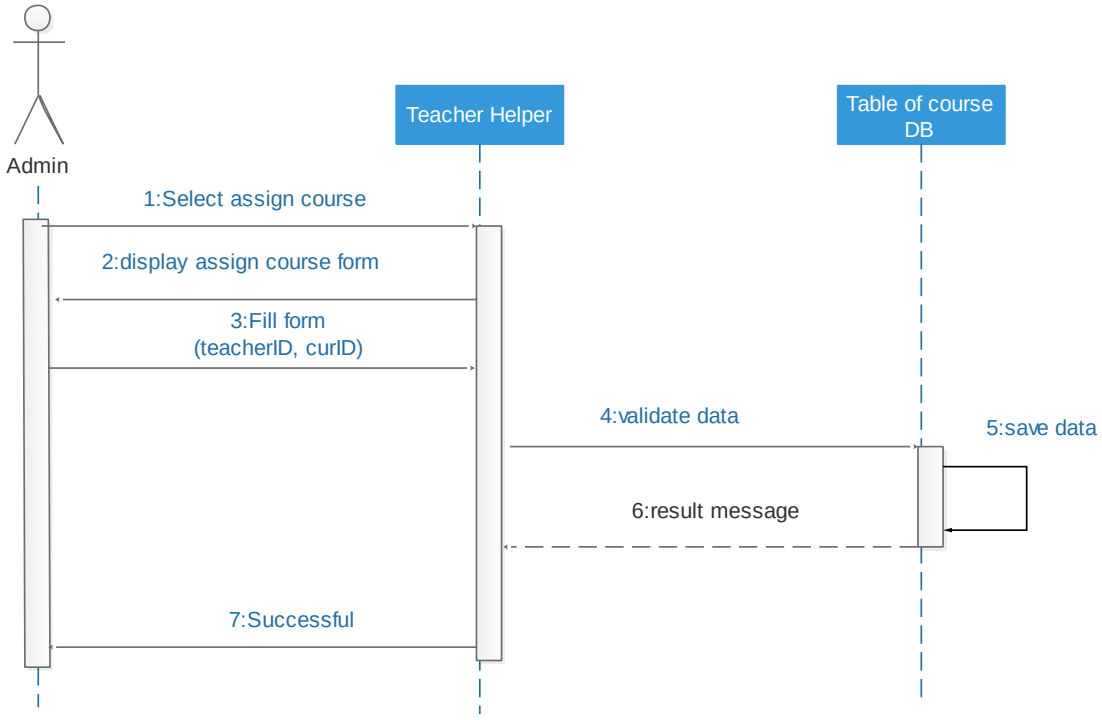


FIGURE 4. 8: Assign course to teacher sequence diagram

4.1.3.3: Record attendance sequence diagram

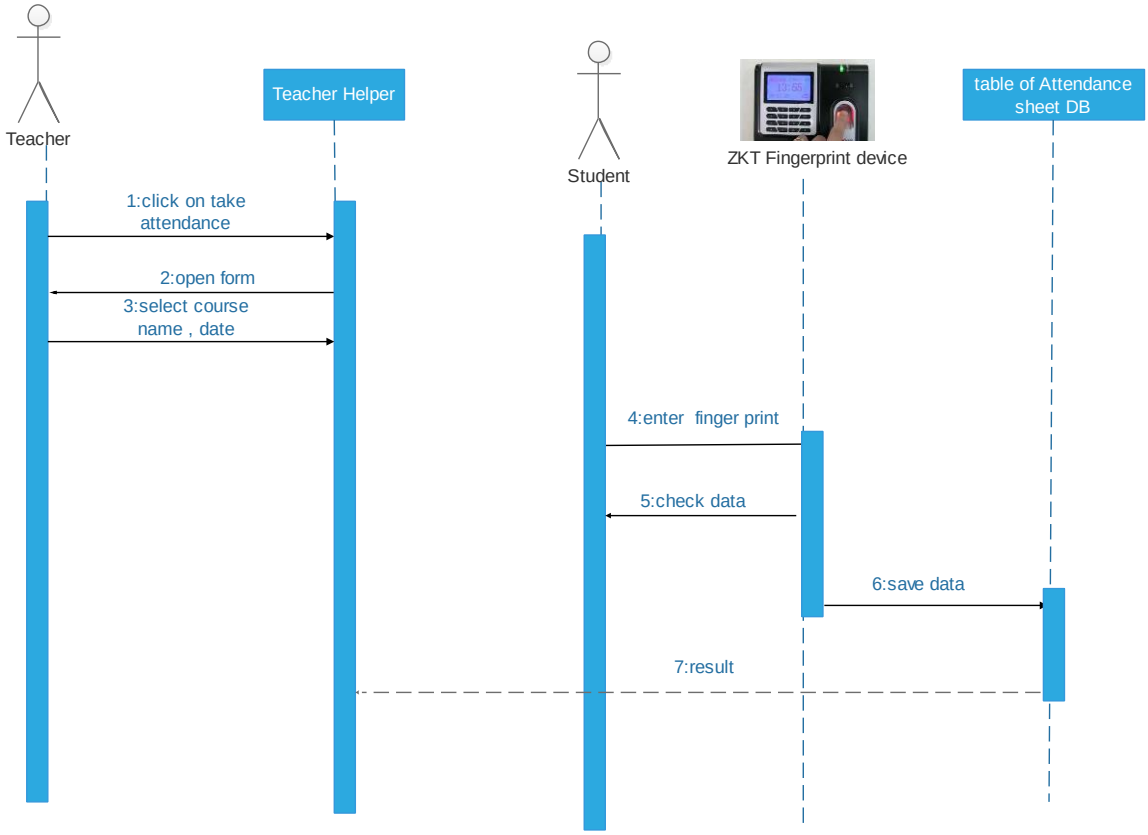


FIGURE 4. 9: Record attendance sequence diagram

4.1.3.4: check absence cause sequence diagram

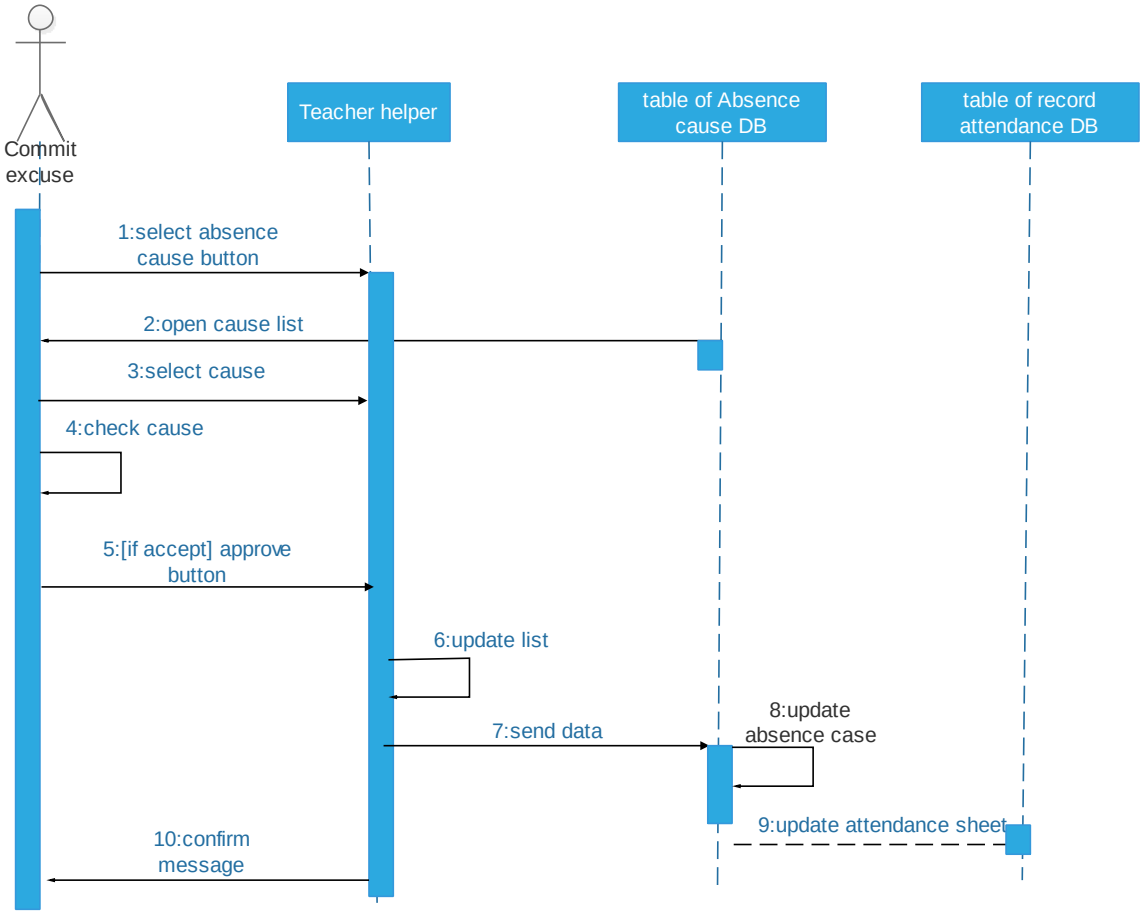


FIGURE 4. 10: check absence cause sequence diagram

4.1.3.5: View attendance sequence diagram

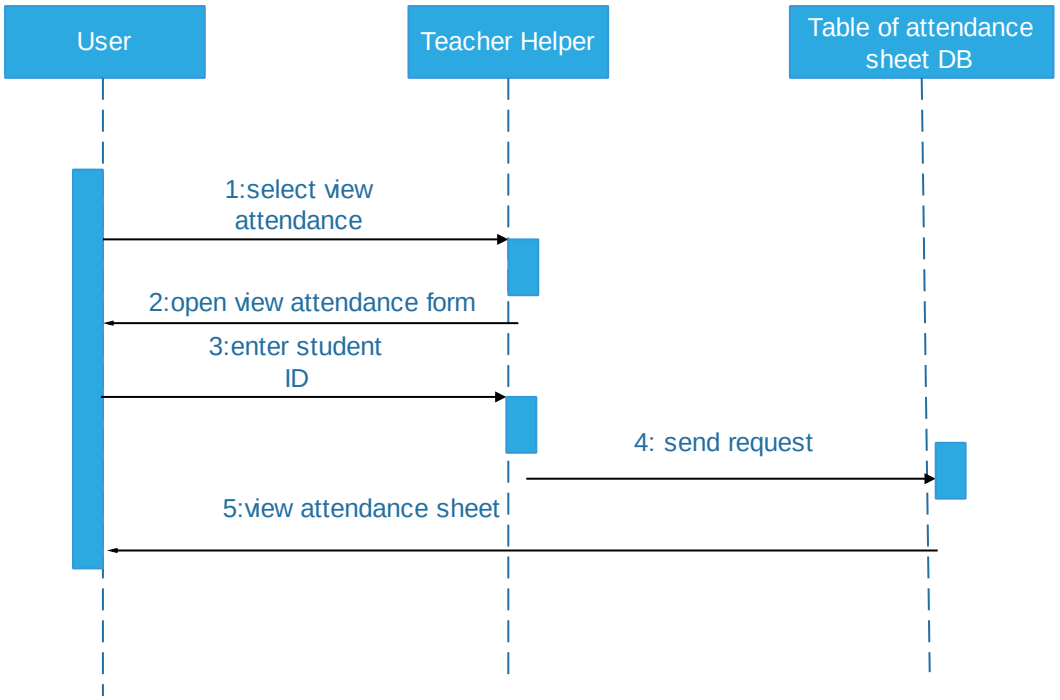


FIGURE 4. 11: View attendance sequence diagram

4.2: Data Modelling

A data model is a description of how data should be used to meet the requirements given by the end user. Data modeling helps to understand the information requirements.

4.2.1: ER Diagram

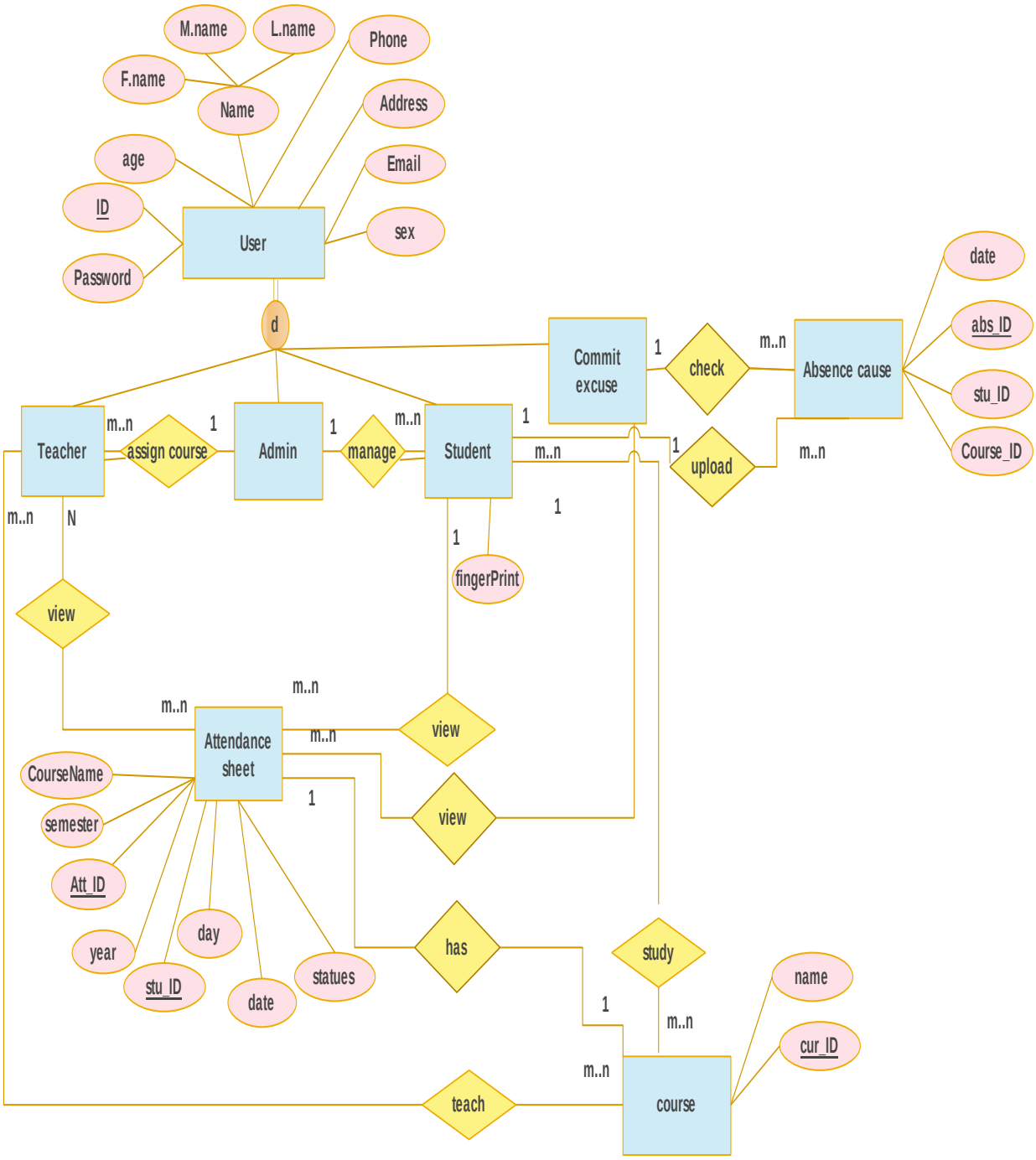


FIGURE 4. 12: ER diagram

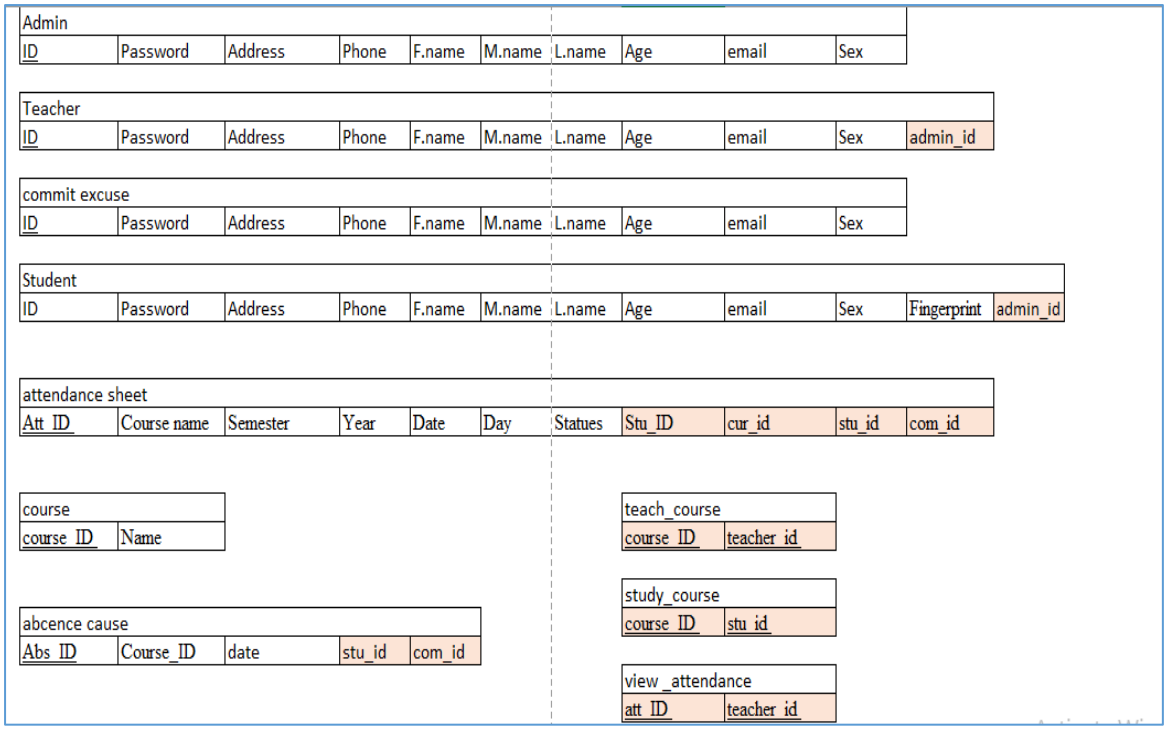


FIGURE 4. 13: Mapping diagram

4.2.2: Data Dictionary

Entity Name	**Multiplicity**	**Relationship**	**Entity Name**	**Multiplicity**
Admin	1	Assign course	Teacher	M..N
Admin	1	Manage	Student	M..N
Admin	1	Upload	Student	M..N
Teacher	M.N	View	Attendance sheet	M..N
Teacher	1	Manage	Attendance sheet	M..N
Student	1	View	Attendance sheet	M..N
Student	1	Upload	Absence cause	M..N
Commit excuse	1	View	Attendance sheet	M..N
Commit excuse	1	Check	Absence cause	M..N
Course	M..N	Study	Student	M..N
Course	M..N	Teach	Teacher	M..N
Course	1	Has	Attendance sheet	1

TABLE 4.1: Data Dictionary showing description of all relationships

Entity Name	Attributes	Description	Data Type	Length	Nulls	Multi-valued	Default	Range	PK
Admin	ID	Uniquely identifies admin	Varchar	20	No	No			Yes
	Password	The password that it used by admin to log in to the teacher helper	Varchar	50	No	No			
	Address	The admin address.	Varchar	50	No	No			
	Phone	The admin phone.	int		No	No			
	F.name	The first name of the admin.	Varchar	20	No	No			
	M.name	The middle name of the admin.	Varchar	20	No	No			
	L.name	The last name of the admin.	Varchar	20	No	No			
	Age	The age of admin.	double		Yes	No			
	email	The email of the admin.	Varchar	50	No	No			
	Sex	Sex of the admin	Varchar	10	No	No			

Teacher	ID	Uniquely identifies teacher	Varchar	20	No	No			Yes
	Password	The password that it used by teacher to log in to the teacher helper	Varchar	50	No	No			
	Address	The teacher address.	Varchar	50	No	No			
	Phone	The teacher phone.	int		No	No			
	F.name	The first name of the teacher.	Varchar	20	No	No			
	M.name	The middle name of the teacher.	Varchar	20	No	No			
	L.name	The last name of the teacher.	Varchar	20	No	No			
	Age	The age of teacher.	double		Yes	No			
	email	The email of the teacher.	Varchar	50	No	No			
	Sex	Sex of the teacher.	Varchar	10	No	No			
Student	ID	Uniquely identifies student	Varchar	20	No	No			Yes

	Password	The password that it used by student to log in to the teacher helper	Varchar	50	No	No			
	Address	The student address.	Varchar	50	No	No			
	Phone	The student phone.	int		No	No			
	F.name	The first name of the student.	Varchar	20	No	No			
	M.name	The middle name of the student.	Varchar	20	No	No			
	L.name	The last name of the student.	Varchar	20	No	No			
	Age	The age of student.	double		Yes	No			
	email	The email of the student.	Varchar	50	No	No			
	Sex	Sex of the student.	Varchar	10	No	No			
	Fingerprint	Fingerprint of student	Int		No	No			
Commit excuse	ID	Uniquely identifies Commit excuse	Varchar	20	No	No			Yes
	Password	The password that it used by	Varchar	50	No	No			

		Commit excuse to log in to the teacher helper							
	Address	The Commit excuse address.	Varchar	50	No	No			
	Phone	The Commit excuse phone.	int		No	No			
	F.name	The first name of the Commit excuse.	Varchar	20	No	No			
	M.name	The middle name of the Commit excuse.	Varchar	20	No	No			
	L.name	The last name of the Commit excuse.	Varchar	20	No	No			
	Age	The age of Commit excuse.	double		Yes	No			
	email	The email of the Commit excuse.	Varchar	50	No	No			
	Sex	Sex of the Commit excuse.	Varchar	10	No	No			
Attendance sheet	Att_ID	Uniquely identifies Attendance sheet	int		No	No			yes
	Course name	The course name of Attendance sheet	Varchar	50	No	No			

	Semester	Semester of Attendance sheet	varchar	20	No	No			
	Date	Date of absence	Date		No	No			
	Day	Day of attendance	Varchar	20	No	No			
	Statues	Statues of attendance	Varchar	20	No	No		Red-green	
	Year	Year of Attendance sheet	Int		No	No			
Course	course_ID	Uniquely identifies course	Varchar	20	No	No			yes
	Name	Name of course	Varchar	20	No	No			
Absence cause	Stu_ID	Uniquely identifies of student	Varchar	20	No	No			Yes
	Abs_ID	Uniquely identifies of cause	Varchar	20	No	No			Yes
	Course_ID	Uniquely identifies of course	Varchar	20	No	No			Yes
	Date	Date of absence	Date		No	No			

TABLE 4.2: Data Dictionary showing description of all attributes

4.3: Detailed Interface Design

Log in:

Function: log in to the application.

Description: Test the ability of the members to log in

Test Input Data: user name, password

Purpose: Login for admin.

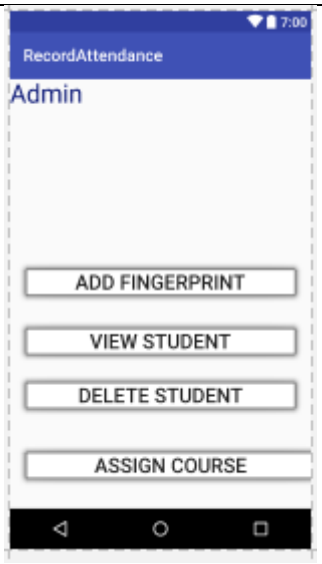
	Step	Action	Input	Expected output	Actual output	Pass/fail
Log in	1	Fill	User name (admin) Password (admin)	Admin control window		Pass
	2	Click login button				

Table 4. 3 log in for admin

Purpose: Login for commit excuse employee.

	Step	Action	Input	Expected output	Actual output	Pass/fail
Log in	1	Fill	User name (emp1) Password (emp1)	employee control window		Pass
	2	Click login button				

Table 4. 4 log in for employee

Purpose: Login for Teacher.

	Step	Action	Input	Expected output	Actual output	Pass/fail
Log in	1	Fill	User name (T_Amal) Password (123)	employee control window		Pass
	2	Click login button				

Table 4. 5 log in for Teacher

Purpose: Login for Student.

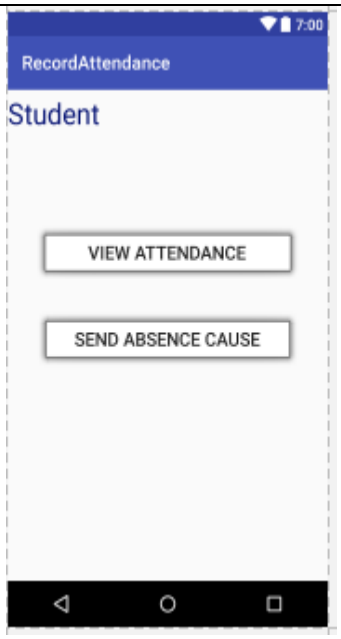
	Step	Action	Input	Expected output	Actual output	Pass/fail
Log in	1	Fill	User name (DH123) Password (123)	employee control window		Pass
	2	Click login button				

Table 4. 6 log in for Student

View Attendance:

Function: view attendance.

Description: Test the ability of the employee to view attendance in application.

Test Input Data: user name, password, student ID

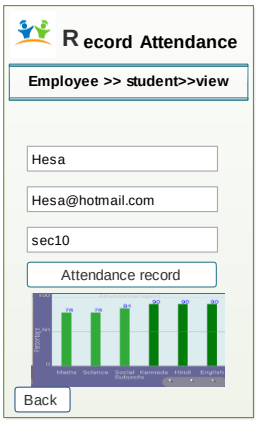
	Step	Action	Input	Expected output	Actual output	Pass/fail
View attendance	1	Fill	User name (emp1) Password (emp1)	Attendance information		Pass
	2	Click login button				
	3	Click view attendance				
	4	Fill	student ID (h1234)			

Table 4.7 view attendance

4.4: Component Diagram

The Component Diagram helps to model the physical aspect of an Object-Oriented software system. It illustrates the architectures of the software components and the dependencies between them. Those software components including run-time components, executable components also the source code components.

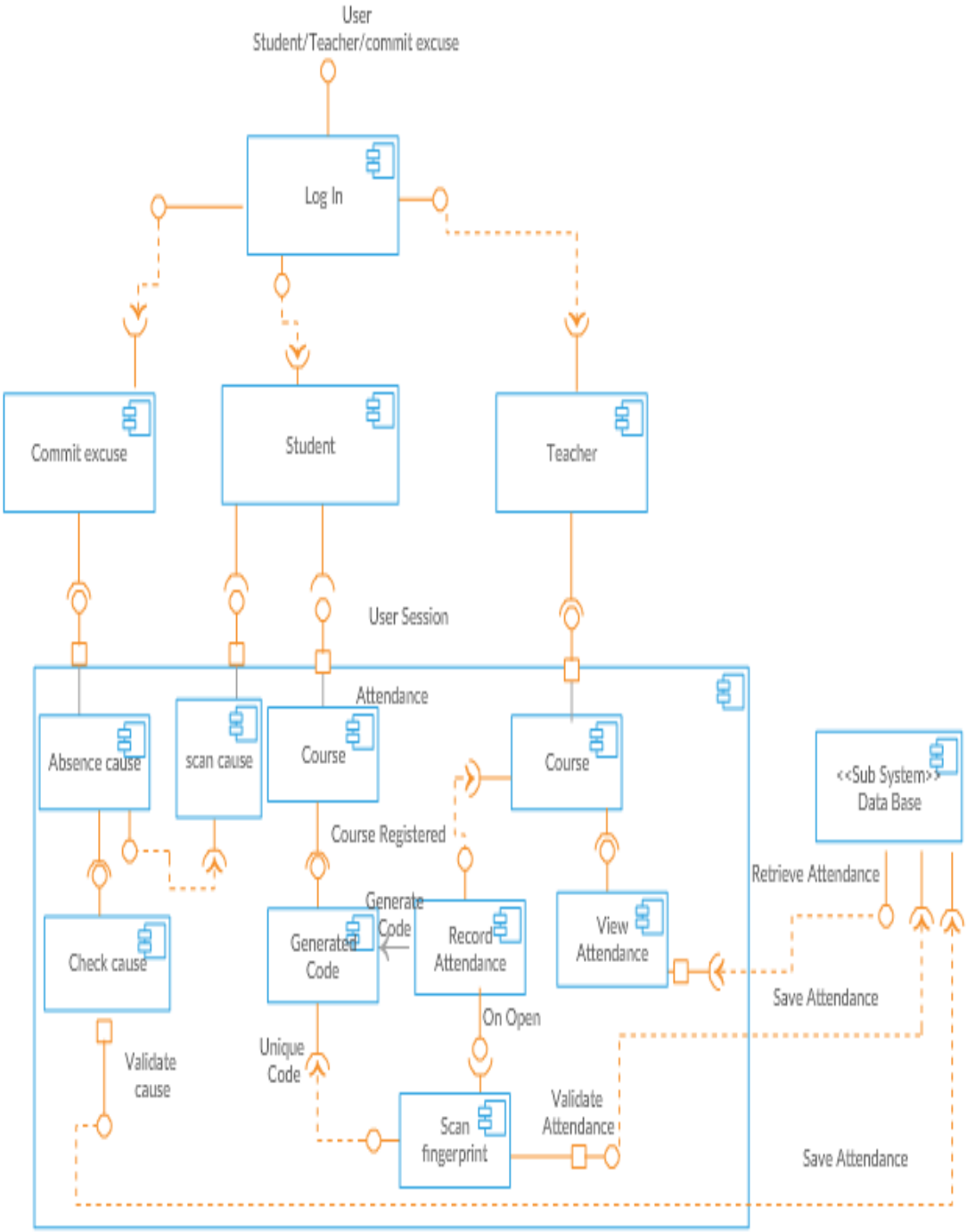


FIGURE 4. 14: Component diagram

4.5: Alternative Designs

We build Teacher Helper system in-house, there are alternative development option such as

- A vendor builds a customized system.
- Buy an existing application and install it
- Lease standard software from an application service provider (ASP), lease as a service (SaaS), or lease via utility computing
- Enter into a partnership or alliance
- Join a third-party e-marketplace or an exchange
- Use a combination of approaches

Chapter five: System Implementation

In this chapter will talk about and explain the hardware, software, deployment in addition to testing in teacher helper application. and how will installation this application.

5.1: System Specification

System specification that contains the hardware and software that use to implement the Teacher Helper application.

5.1.1: Hardware Specifications

This section explains the necessary hardware requirements for using Teacher Helper.
The end user must have a mobile device or tablet supports android application to enable him/her to run the application.

5.1.2: Software Specifications

This section explains the necessary software requirements for using Teacher Helper.
The end user need to have an android system to run the application

5.2: System Testing

It is process of performing a variety of tests on a system to explore functionality or to identify problems. System testing is usually required before and after a system is put in place. The purpose of test is to evaluate the system compliance and achieved requirements. Table 5.1 below show system testing**.**

5.2.1: Features to be tested

The user must enter correct data and fill all data in system

5.2.2: Test Cases

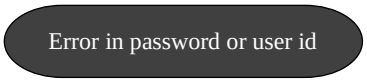
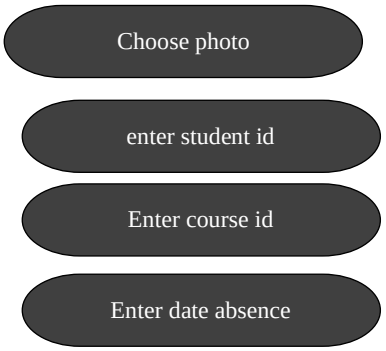
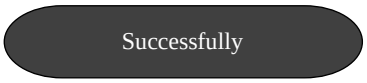
ID	Testing Case	Result
1	- Must enter correct password and user id to login	
2	- The admin must enter student id	
3	– Student must load absence cause image. – Student must enter student id. – Student must enter course id. – Student must date absence.	
4	Alert admin when the assign course process successfully	

Table 5.1 system testing

5.3: System Deployment

This phase comes after the completion of building code and test application functions, it is phase to transfer of program and stat using

5.3.1: Deployment Diagram

This deployment diagram which shows deployment of an application to Android [10].

Android applications are written in Java. Android SDK tools compile and package the code along with any required data and resource files into Android application **archive file** having. apk suffix. The. apk file represents one Android application to be deployed to the Android-enabled mobile devices.

Android applications are composed of one or more application components (activities, services, content providers, and broadcast receivers). Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications).

The manifest (deployment specification) file AndroidManifest.xml describes application requirements, such as the minimum version of Android required and any supported hardware configurations, and it also declares all components in the application.

With Android API Level 8 or later, application could be installed on the **external storage** (for example, on the SD card). This is an optional feature that could be requested for the specific application using a manifest attribute. By default, application is installed on the internal storage of the mobile device and cannot be moved to the external storage.

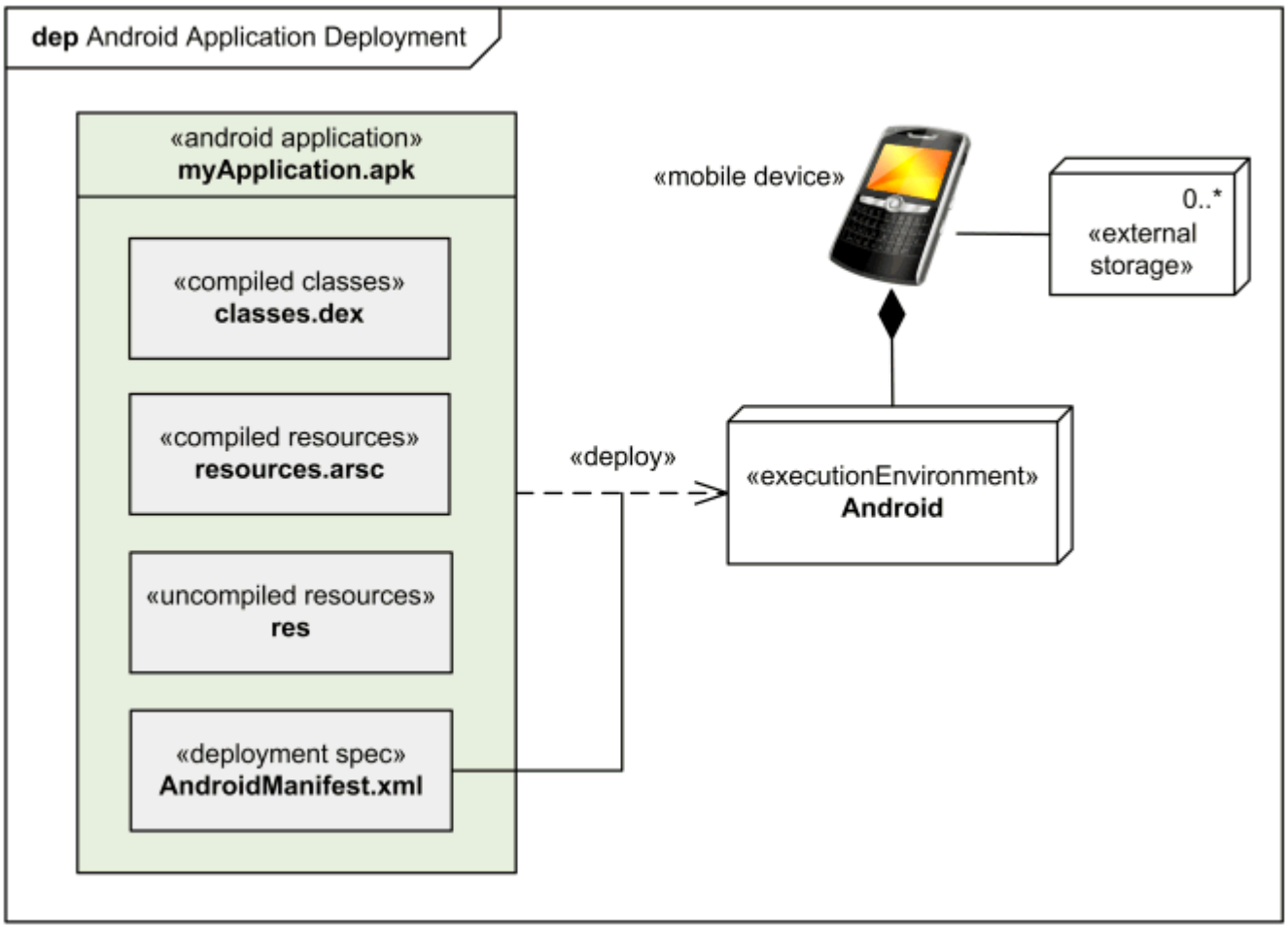


Figure 5.1 Deployment Diagram

5.3.2: Expected Users/Load

This application used by student, teacher, administrator, and employee of commit execute, to help them in management record attendance. Very high application performance because we are using a basic language are Java and MySQL and its servers tolerant millions of users at the same moment also has application interfaces graphical easy so as to be have his performance quickly, but there are other things out of control, which is related to the nature of the device that has the application and also the speed of the internet

5.3.3: Installation Process

To be installing the program on a mobile Android device that requires implementation of the following steps:

1. Should provide a mobile Android system
2. A laptop that contains server and is linked by phone
3. Wireless being linked the phone and the laptop
4. Save the program code format apk
5. Download the apk file on Android phone
6. Android system will create a direct application on the phone
7. Make sure that phone and the laptop is connected to the same Wireless
8. Insert IP address for laptop in the box allocated to it in the mobile because the laptop will work as a servant
9. Now, the mobile phone can be connected to the server and start using.

Conclusion

In this chapter it was presented matters relating to the Deployment of the program and the necessary resources provided, and the cost of these sources, also have been the work of testing the functions of the application

Chapter 6: Conclusion and Future Work

6.1: Conclusion

The main target of Teacher Helper project is to introduce a new service that shall help student and teacher to record attendance and send absence cause. This application helps the student not only to record attendance but also allows them to show attendance sheet of all semester. From a users' perspective, it is now much easier to record attendance. On the other hand, from a commit execute employee' perspective, this application enables them to show absence cause and easy to accept or refuse it. The team members have learned how to work in a group, and the ability to meet the deadlines of a project. Moreover, we have acquired more experience in dealing with different programming tools and techniques.

While developing the application, we faced a lot of difficulties, starting from the difficulty of writing applications in Java, where it is considered a relatively new programming language to us, also how to handle with IED for developing android applications which considers different from another previous developing environment, also how to add fingerprint to database, and how match student fingerprint with that save in database.

These problems were overcome through the organization of a schedule of responsibilities for each member of the team within the different phases of the project. The language of Java was learned through intensive courses online. In addition to learning and work hard on android studio program.
Also, we got a chance to sharpen our skills on developing mobile application in java. We dive into android developing, java and XML, and learn how the application connects to the database to make a robust mobile application.

Also, we bought ZKT fingerprint device to add and match student’s fingerprints.

6.2: Future Work

Even though the system is completely functional and enough to achieve its goals, we will keep it updated by adding more features in the future including:

1) iOS support (due to the high percentage of Apple iPhone's users in Saudi Arabia).
2) Adding Arabic language support (targeting non- English expatriates).
3) Adding barcode scanning method to software.

6.3 Conclusion

In this chapter, we discussed the final thoughts about Teacher Helper project "Teacher Helper" and the objectives achieved. Then we discussed the future work that shall further improve the software.

References

[1] A. dennis, B. Wixom, R. Roth, " system analyses and design edition th5 ", ISBN 978-1-118-05762-9.

[2] Kadry S. et al. (2010). *Wireless Attendance Management System based on Iris Recognition*. Available at: http://www.academicjournals.org/journal/SRE/article-full-text-pdf/29903A917333 [Accessed 11 12 2016].

[3] Reda A. et al. (2011). *A Low-cost Remote Attendance Tracking System for Developing Regions.* Available at: < https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/nsdr011-reda-Hyke.pdf> [Accessed 17 12 2016].

[4] Dhanashree A. G. (2011). *Student Attendance Management.* Available at: < http://www.ijser.org/researchpaper/Student-attendance-software.pdf> [Accessed 21 12 2016].

[5] Microtonics Technologies. (2013, June 4). *RFID Based Attendance Management System.* Available at: < http://www.projectsof8051.com/rfid-based-attendance-management-system > [Accessed 11 12 2016].

[6] Mohammad, A. et al. (2013, December). *Integrated System for Monitoring and Recognizing Students during Class Session.* Available at: < http://airccse.org/journal/jma/5613ijma04.pdf > [Accessed 23 12 2016].

[7] Subramaniam, H. et al. (2013). *Bar Code Scanner Based Student Attendance System (SAS).* Available at: < http://www.researchgate.net/publication/245025631_Bar_Code_Scanner_Based_Student_Attendance_System_(SAS) > [Accessed 24 12 2016].

[8] Talaviya G. et al. (2013). *Wireless Fingerprint Based College Attendance System Using Zigbee Technology*. Available at: < https://www.scribd.com/document/339297671/IA-2015-1304584-1 > [Accessed 24 12 2016].

[9] matching fingerprint," [Online]. Available: https://www.eeweb.com/blog/ti/id-systems-for-smarter-biometrics [Accessed 24 4 2017].

[10] Deployment diagram," [Online]. Available: http://www.uml-diagrams.org [Accessed 24 4 2017].

Appendix

A. Code Snippets

```
package com.dh.me.recordattendance.ui;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.Toast;
import com.dh.me.recordattendance.Helper.WebServiceHelper.ApiEndPoint;
import com.dh.me.recordattendance.Helper.WebServiceHelper.ApiEndpointInterface;
import com.dh.me.recordattendance.R;
import com.google.gson.JsonObject;
import retrofit2.Call;
import retrofit2.Callback;
```

```
public class Home extends AppCompatActivity {
    // declare variable
    public EditText userid1, password1,userid2,password2,userid3,password3,userid4,password4;
    Activity mContext;
   private Button btnStudent, btnAdmin,btnEmp,btnTeacher;
    public String  login_name1,   login_password1;
    public String  login_name2,   login_password2;
    public String  login_name3,   login_password3;
    public String  login_name4,   login_password4;
    public static   String loginId;
    public static   int loginType;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_home);
       mContext=this;
       //connect variable with tools in layout by id of tools
       userid1 = (EditText) findViewById(R.id.editText_userid);
       password1 = (EditText) findViewById(R.id.editText_password);
       userid2 = (EditText) findViewById(R.id.emp_userid);
       password2 = (EditText) findViewById(R.id.emp_password);
       userid3 = (EditText) findViewById(R.id.teacher_userid);
       password3 = (EditText) findViewById(R.id.teacher_password);
       userid4 = (EditText) findViewById(R.id.student_userid);
       password4 = (EditText) findViewById(R.id.student_password);
       btnStudent = (Button) findViewById(R.id.btn_login_student);
       btnStudent.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            loginStudent( login_name4,login_password4);
        }
    });
    btnAdmin = (Button) findViewById(R.id.btn_login);
    btnAdmin.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            loginAdmin( login_name1,login_password1);
        }
    });
    btnEmp = (Button) findViewById(R.id.btn_login_emp);
    btnEmp.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            loginEmp( login_name2,login_password2);
        }
    });
    btnTeacher = (Button) findViewById(R.id.btn_login_teacher);
    btnTeacher.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            loginTeacher( login_name3,login_password3);
        }
    });
```

```
// select item menu
 @Override
 public boolean onCreateOptionsMenu(Menu menu) {
     MenuInflater inflater = getMenuInflater();
     inflater.inflate(R.menu.user_type, menu);
    // return true;
     return super.onCreateOptionsMenu(menu);
 }

 @Override
 public boolean onOptionsItemSelected(MenuItem item) {
   //declare relative layout

     RelativeLayout radmin = (RelativeLayout) findViewById(R.id.rl_admin);
     RelativeLayout remp = (RelativeLayout) findViewById(R.id.rl_emp);
     RelativeLayout rstu = (RelativeLayout) findViewById(R.id.rl_student);
     RelativeLayout rteah = (RelativeLayout) findViewById(R.id.rl_teacher);
```

```
switch (item.getItemId()) {
    case R.id.admin:
        if (item.isChecked()) item.setChecked(false);
        else item.setChecked(true);
        radmin.setVisibility(View.VISIBLE);
        remp.setVisibility(View.INVISIBLE);
        rstu.setVisibility(View.INVISIBLE);
        rteah.setVisibility(View.INVISIBLE);
        return true;
    case R.id.emp:
        if (item.isChecked()) item.setChecked(false);
        else item.setChecked(true);
        radmin.setVisibility(View.INVISIBLE);
        rstu.setVisibility(View.INVISIBLE);
        rteah.setVisibility(View.INVISIBLE);
        remp.setVisibility(View.VISIBLE);
        return true;
    case R.id.student:
        if (item.isChecked()) item.setChecked(false);
        else item.setChecked(true);
        remp.setVisibility(View.INVISIBLE);
        radmin.setVisibility(View.INVISIBLE);
        rteah.setVisibility(View.INVISIBLE);
        rstu.setVisibility(View.VISIBLE);
        return true;
    case R.id.teacher:
        if (item.isChecked()) item.setChecked(false);
        else item.setChecked(true);
        remp.setVisibility(View.INVISIBLE);
```

```
        case R.id.teacher:
            if (item.isChecked()) item.setChecked(false);
            else item.setChecked(true);
            remp.setVisibility(View.INVISIBLE);
            rstu.setVisibility(View.INVISIBLE);
            radmin.setVisibility(View.INVISIBLE);
            rteah.setVisibility(View.VISIBLE);
            return true;
        default:
    }
    return super.onOptionsItemSelected(item);
}
```

```
// code of admin log in button
private void loginAdmin( String login_name1,  String login_password1)
{
    login_name1 = userid1.getText().toString().trim();
    login_password1 = password1.getText().toString().trim();

  final   ProgressDialog loading = ProgressDialog.show(Home.this, "pleas wait..", "..", false, false)
    loginId=login_name1;
    loginType=0;
    ApiEndpointInterface objApiEndpointInterface = ApiEndPoint.getClient().create(ApiEndpointInterfac

    Call<JsonObject> objCall = objApiEndpointInterface.loginAdmin( login_name1,login_password1);

    objCall.enqueue(new Callback<JsonObject>()
    {
        @Override
        public void onResponse(Call<JsonObject> call, retrofit2.Response<JsonObject> response) {
            loading.dismiss();
            try {
                String result = response.body().get("success").toString();
                if (result.equals("1")){
                    Toast.makeText(mContext,"correct ",Toast.LENGTH_LONG).show();
                    startActivity(new Intent(mContext,adminHome.class));
                }else {
                    Toast.makeText(Home.this, "Error in password or id ", Toast.LENGTH_LONG).show();
                }
            } catch (Exception objException) {
                Toast.makeText(Home.this, "Error in Server", Toast.LENGTH_LONG).show();

            }

            ƒ


            @Override
            public void onFailure(Call<JsonObject> call, Throwable t)
            {

                loading.dismiss();
                String r = "";
            }


    });
}
```

```
// code of teacher log in button
private void loginTeacher( String login_name3,  String login_password3)
{
    login_name3 = userid3.getText().toString().trim();
    login_password3 = password3.getText().toString().trim();

    final   ProgressDialog loading = ProgressDialog.show(Home.this, "pleas wait..", "..", false, false);

    loginId=login_name3;
    loginType=2;
    ApiEndpointInterface objApiEndpointInterface = ApiEndPoint.getClient().create(ApiEndpointInterface.class);
    Call<JsonObject> objCall = objApiEndpointInterface.loginTeacher( login_name3,login_password3);


    objCall.enqueue(new Callback<JsonObject>() {
        @Override
        public void onResponse(Call<JsonObject> call, retrofit2.Response<JsonObject> response) {

            loading.dismiss();
            try {
                String result = response.body().get("success").toString();
                if (result.equals("1")){
                    Toast.makeText(mContext,"correct ",Toast.LENGTH_LONG).show();
                    startActivity(new Intent(mContext,teacherHome.class));
                }else {
                    Toast.makeText(Home.this, "Error in password or id ", Toast.LENGTH_LONG).show();
                }
            } catch (Exception objException) {
                Toast.makeText(Home.this, "Error in Server", Toast.LENGTH_LONG).show();
```

```
// code of stusent log in button
private void loginStudent( String login_name4,  String login_password4)
{
    login_name4 = userid4.getText().toString().trim();
     login_password4 = password4.getText().toString().trim();
   ApiEndpointInterface objApiEndpointInterface = ApiEndPoint.getClient().create(ApiEndpointInterface.class);
   Call<JsonObject> objCall = objApiEndpointInterface.loginStudent( login_name4,login_password4);
   loginId=login_name4;
   loginType=3;
   final   ProgressDialog loading = ProgressDialog.show(Home.this, "pleas wait..", "..", false, false);
   objCall.enqueue(new Callback<JsonObject>() {
        @Override
        public void onResponse(Call<JsonObject> call, retrofit2.Response<JsonObject> response) {
            loading.dismiss();
            try {
                String result = response.body().get("success").toString();
                if (result.equals("1")){
                    Toast.makeText(mContext,"correct ",Toast.LENGTH_LONG).show();
                    startActivity(new Intent(mContext,studentHome.class));
                }else {
                    Toast.makeText(Home.this, "Error in password or id ", Toast.LENGTH_LONG).show();
                }
            } catch (Exception objException) {
                Toast.makeText(Home.this, "Error in Server", Toast.LENGTH_LONG).show();
            }
        }


        @Override
        public void onFailure(Call<JsonObject> call, Throwable t) {
```

```
        }


        @Override
        public void onFailure(Call<JsonObject> call, Throwable t) {
            loading.dismiss();


            String r = "";
        }


    });
}


}
```

Appendix A.1 home page code

```
package com.dh.me.recordattendance.ui;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import com.dh.me.recordattendance.R;
public class adminHome extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_home);
    }
    public void btn_addfingerprint(View v)
    {
        startActivity(new Intent(adminHome.this,AddStudentFinger.class));
    }
    public void btn_viewstudent(View v)
    {
        startActivity(new Intent(adminHome.this,viewStudent.class));
    }

    public void btn_deletetudent(View v)
    {
        startActivity(new Intent(adminHome.this,deleteStudent.class));
    }
     public void btn_assign(View v)
    {

        startActivity(new Intent(empHome.this,assignCourse.class));
```

Appendix A.2 Admin home page code

```
private void assignCours(){
    final String cid = coursid.getText().toString().trim();
    final String tid = teacherid.getText().toString().trim();
    StringRequest stringRequest = new StringRequest(Request.Method.POST, AppConfigHelper.BASE_URL+"assignCourse.php",
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    Toast.makeText(assignCourse.this,response,Toast.LENGTH_LONG).show();
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {

                  Toast.makeText(assignCourse.this,error.toString(),Toast.LENGTH_LONG).show();
                }
            }){
        @Override
        protected Map<String,String> getParams(){
            HashMap<String, String> params = new HashMap<String, String>();
            params.put("C_id",cid);
            params.put("T_id",tid);

            return params;
        }
    };
```

Appendix A.3 assign course to teacher code

```
public class viewAttendanceTeacher extends AppCompatActivity {
    Button btn_viewatt,btn_viewattweakly;
    EditText editText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_attendance_teacher);

        btn_viewatt=(Button)findViewById(R.id.btn_viewatt);

        btn_viewattweakly=(Button)findViewById(R.id.btn_viewattweakly);
        editText=(EditText) findViewById(R.id.editText);

        btn_viewatt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Calendar c = Calendar.getInstance();

                int year = c.get(Calendar.YEAR);
                int month = c.get(Calendar.MONTH)+1;
                int day = c.get(Calendar.DAY_OF_MONTH);

String id="0";
            if (!TextUtils.isEmpty( editText.getText().toString()))
                {
                    id=editText.getText().toString();

                }
```

```
            Intent i = new Intent(viewAttendanceTeacher.this, viewAttendancereport.class);
            i.putExtra("id",id);
            i.putExtra("day",day);
            i.putExtra("month",month);
            i.putExtra("year",year);
            i.putExtra("curid","0");
            i.putExtra("t",Home.loginId);
            viewAttendanceTeacher.this.startActivity(i);
        }
    });

    btn_viewattweakly.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Calendar c = Calendar.getInstance();
            int year = c.get(Calendar.YEAR);
            int month = c.get(Calendar.MONTH)+1;
            int day = c.get(Calendar.DAY_OF_MONTH);
            Intent i = new Intent(viewAttendanceTeacher.this, viewAttendancereportchart.class);
            String id="0";
            if (!TextUtils.isEmpty( editText.getText().toString()))
            {
                id=editText.getText().toString();

            }
            i.putExtra("id",id);
            i.putExtra("day",day+"");
            i.putExtra("month",month+"");
            i.putExtra("year",year+"");
```

```
                i.putExtra("year",year+"");
                i.putExtra("curid","0");
                i.putExtra("t",Home.loginId);
                viewAttendanceTeacher.this.startActivity(i);
            }
        });

    }
}
```

Appendix A.4 teacher view attendance code

```php
<?php
include("connectdb.php");
$name=$_POST["Login_name"];
$pass=$_POST["Login_password"];

$sql = "select * from commit_excuse where c_id='$name' and password='$pass'";

$result = $conn->query($sql);

if ($result->num_rows > 0)
    {

$check["success"] =1;
$check["message"]="correct";

    }
else
{
    $check["success"] =0;
    $check["message"]="Invalid user name or password, please try again";
}
echo json_encode($check);
$conn->close();

?>
```

Appendix A.5 query for retrieve data from commit excuse table to employee login

```php
<?php
include("connectdb.php");
$cid=$_POST["C_id"];
$sid=$_POST["S_id"];
$da=$_POST["DateA"];
$eimg=$_POST["photo"];
$emp="emp1";
$check=array();
$file_path = "image/";
$dt=date("Ymd"); //date
$tm=date("his"); //time
$check["message"]="";
$rnd = rand(0, 5000);
$imgname=basename( $_FILES['uploaded_file']['name']);

    $file_path = $file_path . basename( $_FILES['uploaded_file']['name']);
    if(move_uploaded_file($_FILES['uploaded_file']['tmp_name'], $file_path)) {
        }
      $sql = "INSERT INTO absence_cause(date, s_id, cur_id, c_id, cause) VALUES ('$da','$sid','$cid','$emp','$imgname')";
$result = $conn->query($sql);
if ($result) {
        $check["message"]="Successfully ";
}
else
{
    $check["message"]="error".mysqli_error();
}
echo json_encode($check);
$conn->close();
?>
```

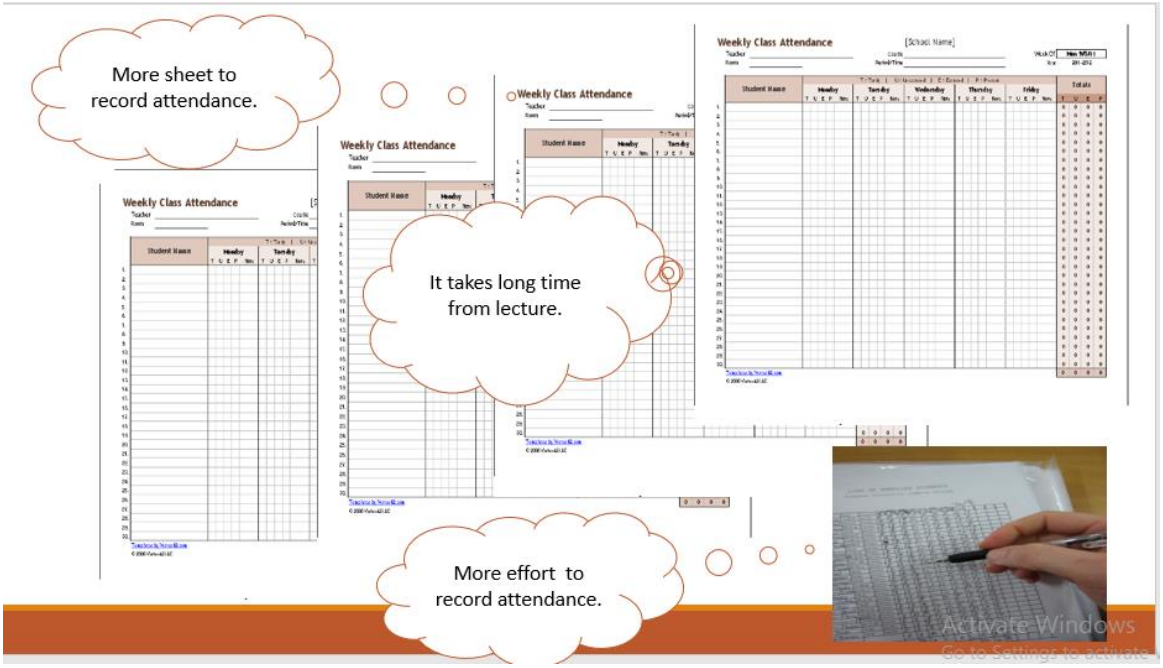

Appendix A.6query for upload absence cause

```php
<?php
include("connectdb.php");
$id= $_POST['id'];
$day= $_POST['day'];
$month= $_POST['month'];
$year= $_POST['year'];
$response=array();
$sql = "SELECT DISTINCT
    `attendance_sheet`.`att_id`
    , `attendance_sheet`.`date`
    , `attendance_sheet`.`course_name`
    , `attendance_sheet`.`cur_id`
    ,CONCAT( `student`.`f_name`,' ',
    `student`.`m_name`,' '
    , `student`.`l_name`) AS name,`attendance_sheet`.`s_id` as id
    , `attendance_sheet`.`statues` as status
FROM
    `attendance_sheet`
    INNER JOIN `student`
        ON (`attendance_sheet`.`s_id` = `student`.`s_id`)
WHERE  `attendance_sheet`.`cur_id`='$id' and (DAY(`date`)='$day'

    AND  YEAR(`date`)='$year' AND MONTH(`date`)='$month')
        ;";
$r = mysqli_query($conn,$sql);
$response["data"]=array();
while($row = mysqli_fetch_assoc($r))
        {
    $result = array();
$result[]=$row;
```

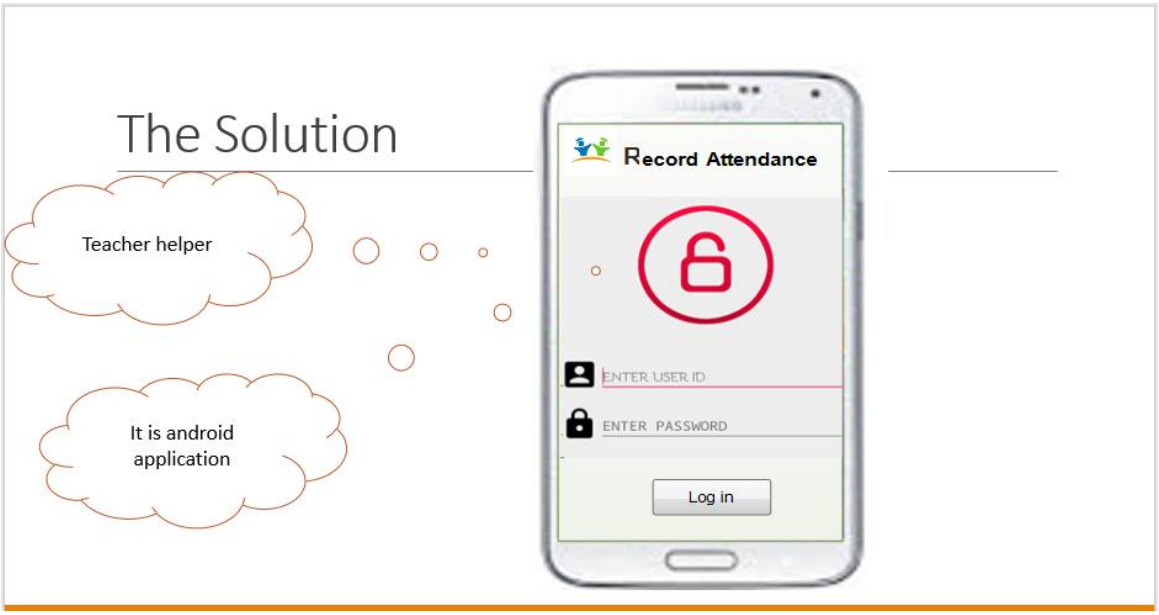
```php
$result[]=$row;

        array_push($response["data"], $row);
        }
                $response["success"] = 1;
        echo json_encode($response);
  $conn->close();
?>
```

Appendix A.7 query for view attendance of student

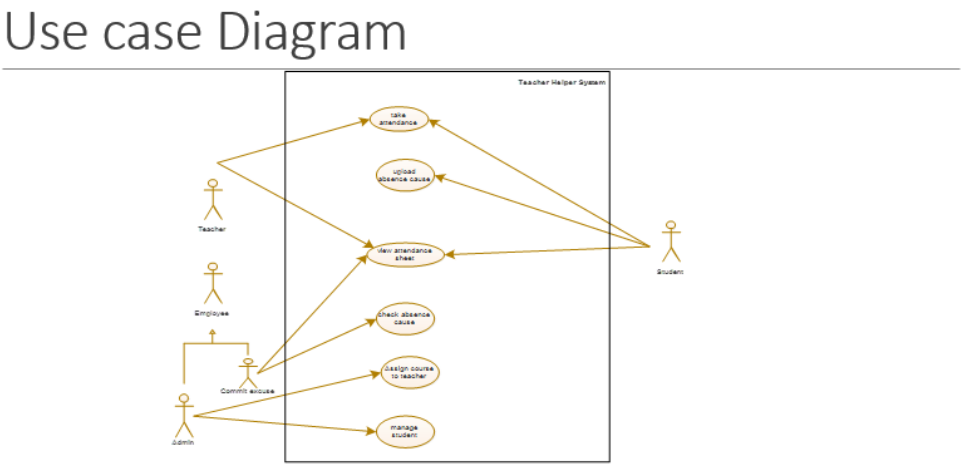
B. Presentation Slides



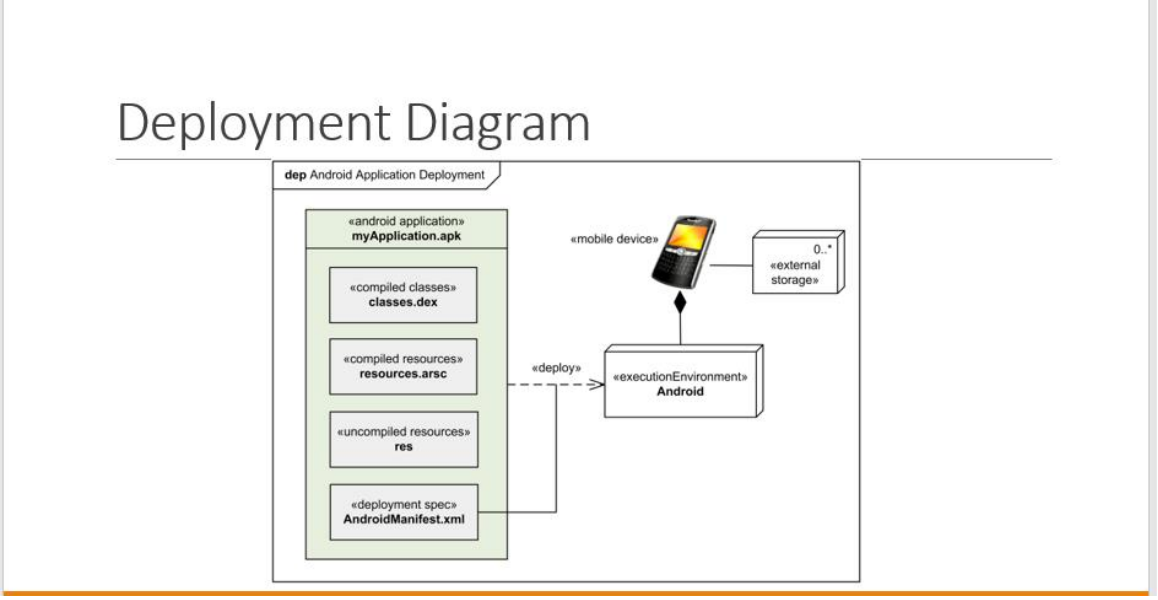
Appendix B.1 problem slide presentation



Appendix B.2 solution slide presentation



Appendix B.3 use case slide presentation



Appendix B.4deployment slide presentation

C. Miscellaneous

Teacher helper

we are a student in Al-Imam Mohammad ibn Saud Islamic University, college of Computer and Information Sciences, Information Systems Department. We show questionnaire about our graduate project. Teacher helper is mobile application works on the android operating system. Teacher Helper *project aims to manage the attendance of students, presenting absence causing and saving time and effort of teachers and students.*

1. Do you have any application to record attendance students?
   - Yes
   - No
2. How do you record attendance student?
   - By calling student name
   - By signature of students
   - Other
3. Do this way waste lecture time?
   - Yes
   - No
4. Is there a communication way between teacher and commit excuse?
   - Yes
   - No
5. what is the way of send absent excuse to the teacher?
   - By phone
   - by manually
   - By Email
6. Is there a way to knowledge if student exceeded the number of allowable absences?
   - Yes

- No

7. does commit excuse send an alarm message to student exceeded the number of allowable absences?
   - Yes
   - No
8. Do you see this application will save the time of teacher and employee?
   - Yes
   - No
   - Maybe

نحن طالبات جامعة الأمام محمد بن سعود الاسلامية من كلية علوم الحاسب و المعلومات قسم نظم معلومات . هذا الاستبيان يستهدف مشروعنا وهو عبارة عن تطبيق يسهل تسجيل حضور الطلبه داخل المحاضرة من خلال بصمة الأصبع أو الباركود , وهو يسهل على الطلبة اطلاعهم على سجل الحضور ومعرفة عدد ساعاتهم المتبقيه وارفاق العذر بشكل تقني وارسال تنبيه للطلبة قبل تجاوز الحرمان من المادة و يساعد المعلم على أدارة الحضور والحفاظ على الوقت , وهذا التطبيق يسهل التواصل بين الطلبة و المعلم ولجنة الاعذار.

١. هل يوجد أي تطبيق لتسجيل حضور الطلاب؟
   a. نعم
   b. لا
٢. كيف يتم تسجيل حضور الطلاب؟
   a. بالنداء على اسماء الطلاب
   b. بتوقيع الطلاب في كشف الحضور
   c. الاثنين معا
٣. هل هذه الطريقة تضيع الوقت؟
   a. نعم
   b. لا
٤. هل يوجد طريقة إتصال بين لجنة الأعذار والمعلمين؟
   a. نعم
   b. لا
٥. ما هي طريقة إرسال عذر الغياب
   a. يدويا
   b. بالتليفون
   c. بالايميل
٦. هل يوجد طريقة لمعرفة ان كان الطالب قارب على تجاوز نسبة الغياب المسموح بها؟
   a. نعم
   b. لا
٧. هل لجنة الأعذار ترسل رسالة تنبيهية للطلاب الذين قاربوا على تجاوز نسبة الغياب المسموح بها؟
   a. نعم
   b. لا
٨. هل ترى أن هذا التطبيق سيوفر الوقت للطلاب وللمدرسين وموظفي لجنة الأعذار؟
   a. نعم
   b. لا
   c. ربما

APPENDEX B

نحن طالبات جامعة الأمام محمد بن سعود الاسلامية من كلية علوم الحاسب و المعلومات قسم نظم معلومات . هذا الاستبيان يستهدف مشروعنا وهو عبارة عن تطبيق يسهل تسجيل حضور الطلبه داخل المحاضرة من خلال بصمة الأصبع أو الباركود , وهو يسهل على الطلبة اطلاعهم على سجل الحضور ومعرفة عدد ساعاتهم المتبقيه وارفاق العذر بشكل تقني وارسال تنبيه للطلبة قبل تجاوز الحرمان من المادة و يساعد المعلم على أدارة الحضور والحفاظ على الوقت , وهذا التطبيق يسهل التواصل بين الطلبة و المعلم ولجنة الاعذار.

١. هل تستطيع الاطلاع على ملف حضورك طول الترم لكل مادة ؟ *
   a. نعم
   b. لا
٢. هل تواجه صعوبة في تسليم عذر الغياب للجنة الأعذار؟ *
   a. نعم
   b. لا
   c. أحيانا
٣. هل يوجد طريقة لمعرفة إن تم قبول عذر غيابك أم لا ؟ *
   a. نعم
   b. لا
٤. هل توجد طريقة اخرى غير الذهاب للمعلم لمعرفة عدد الساعات المتبقية ؟ *
   a. نعم
   b. لا
٥. اذا أجابتك بـ لا هل ترى أن التطبيق يساعدك في معرفة عدد ساعاتك ؟ *
   a. نعم
   b. لا
٦. هل يتم اخبارك من قبل الكلية أنك تجاوزت عدد مرات الغياب؟ *
   a. نعم
   b. لا
٧. هل تفضل استخدام التطبيق لتسجيل الحضور في المحاضرة ؟ *
   a. نعم
   b. لا
٨. هل تفضل استخدام موبايلك لتسجيل حضورك ؟ *
   a. نعم
   b. لا
٩. ما هي الطريقة التي تفضلها لتسجيل الحضور الكتروني ؟ *
   a. بصمة الأصبع
   b. الباركود
   c. الأثنين معا
١٠. هل ترى أن هذا التطبيق سوف يسهل عليك التواصل مع المعلم و لجنة الأعذار ؟ *
   a. نعم
   b. لا
   c. أحيانا
١١. هل لديك أي اقترحات أخرى ؟
   a. أخرى:

APPENDEX C

Result of student questionnaire:

FIGURE C.1 Result of student questionnaire Q1

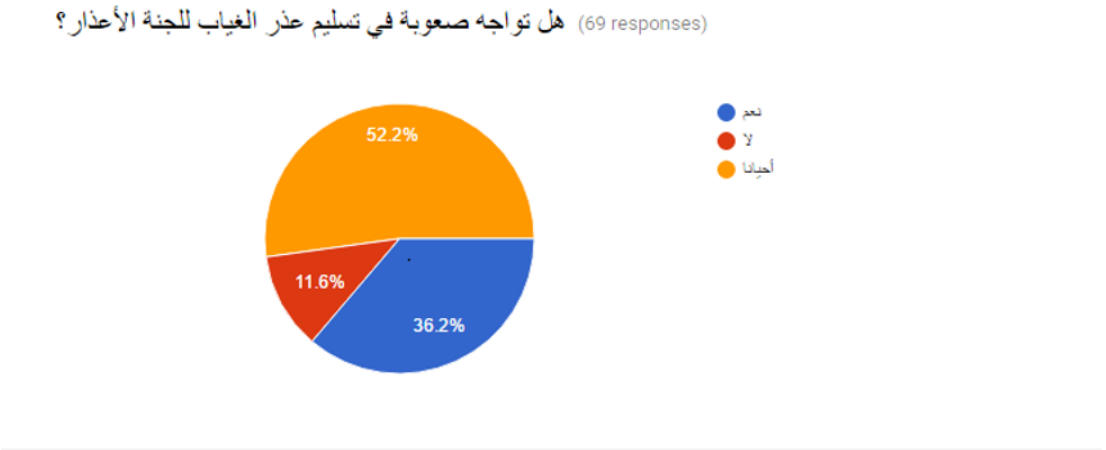


FIGURE C.2 Result of student questionnaire Q2

FIGURE C.3 Result of student questionnaire Q3

FIGURE C.4 Result of student questionnaire Q4

FIGURE C.5 Result of student questionnaire Q5

هل يتم اخبارك من قبل الكلية أنك تجاوزت عدد مرات الغياب؟ (69 responses)

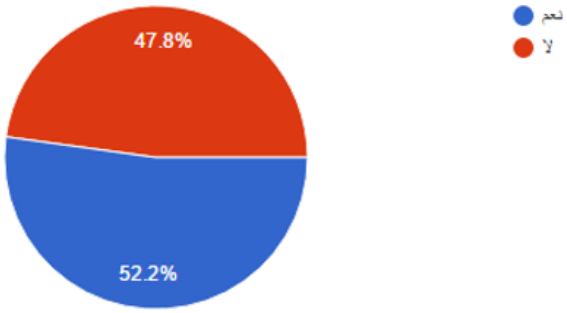


FIGURE C.6 Result of student questionnaire Q6

هل تفضل استخدام التطبيق لتسجيل الحضور في المحاضرة ؟ (69 responses)

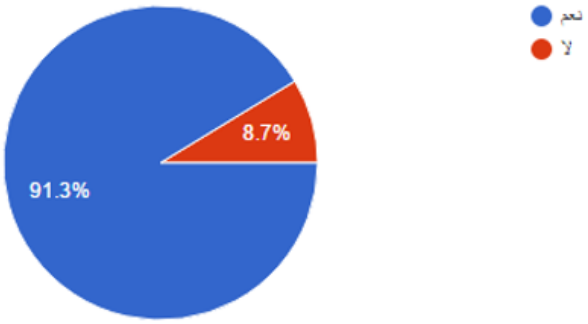


FIGURE C.7 Result of student questionnaire Q7

هل تفضل استخدام موبايلك لتسجيل حضورك ؟ (69 responses)

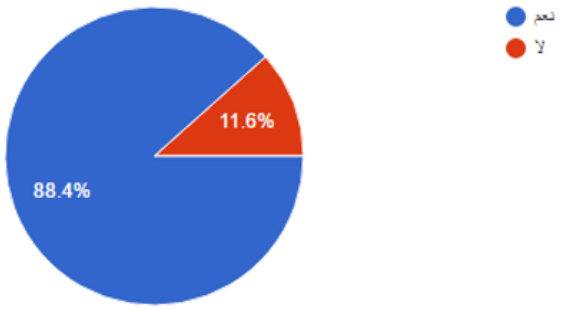


FIGURE C.8 Result of student questionnaire Q8

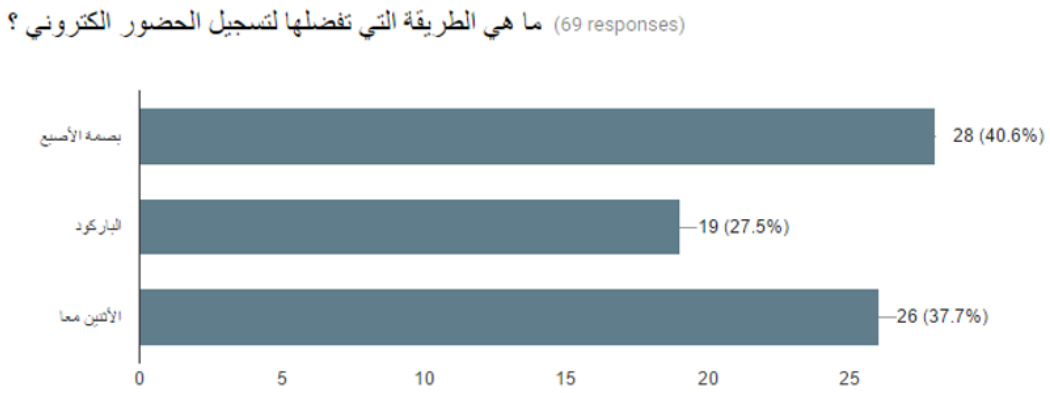


FIGURE C.9 Result of student questionnaire Q9

FIGURE C.10 Result of student questionnaire Q10

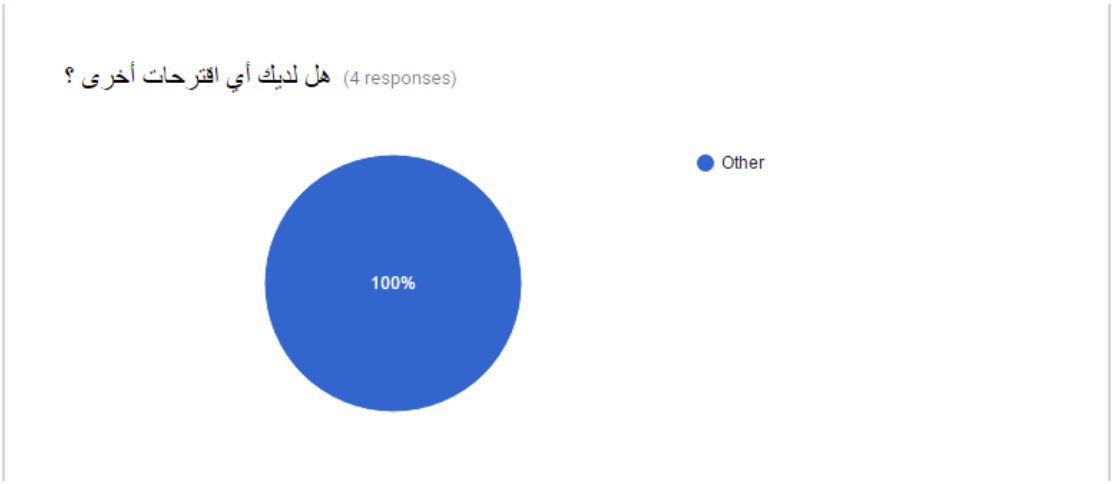


FIGURE C.11 Result of student questionnaire Q11

Result of teacher’s questionnaire

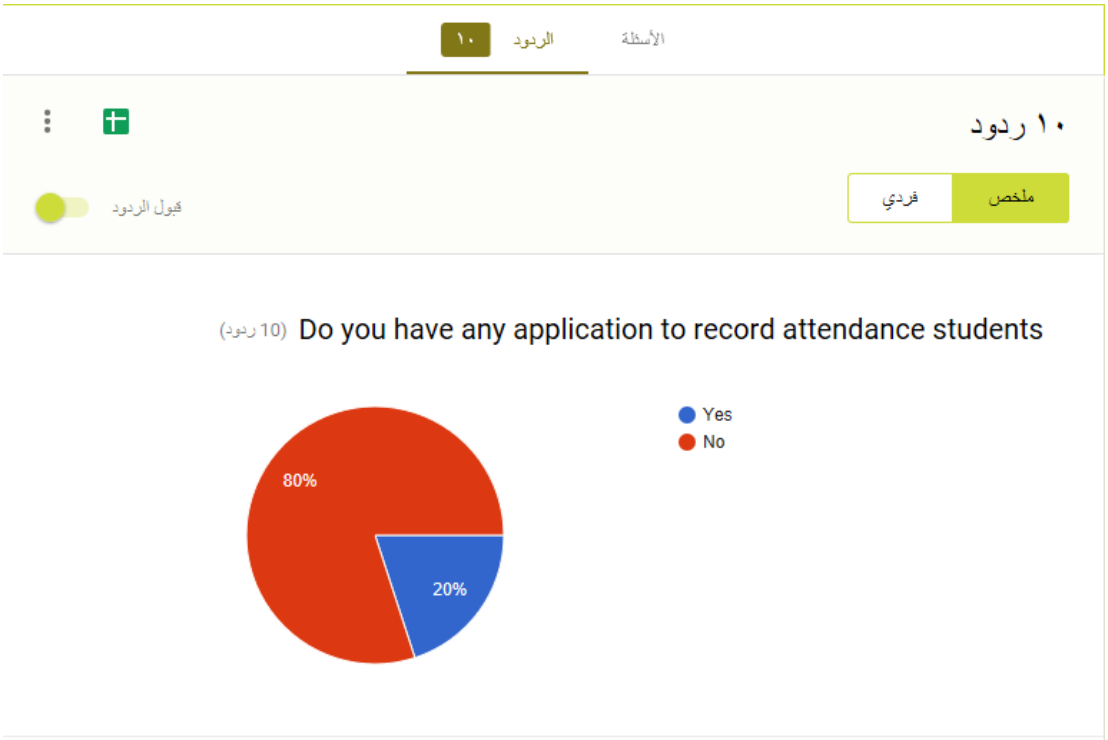


FIGURE C.12 Result of teacher’s questionnaire Q1

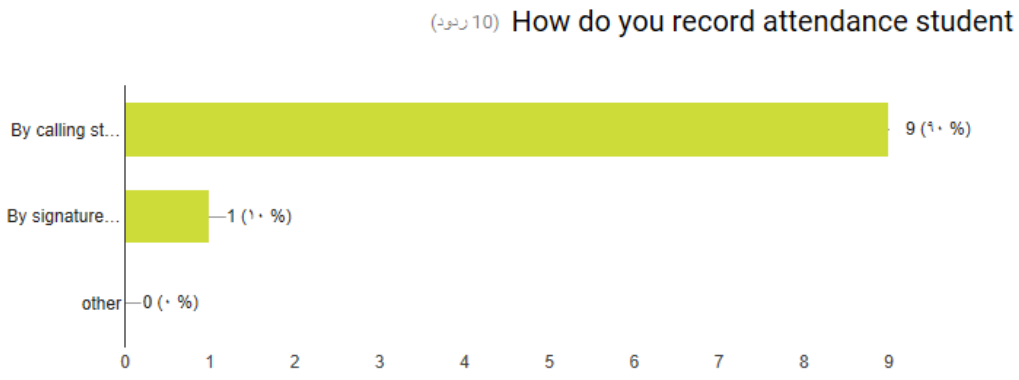


FIGURE C.13 Result of teacher's questionnaire Q2

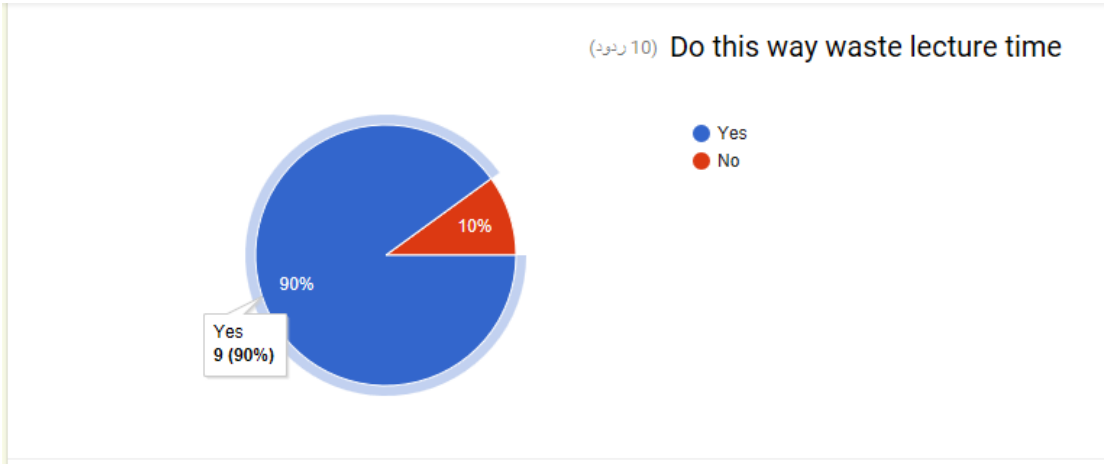


FIGURE C.14 Result of teacher's questionnaire Q3

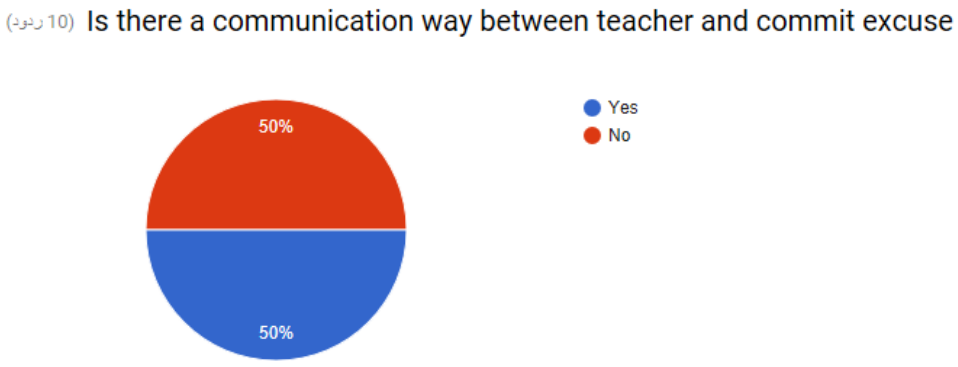


FIGURE C.15 Result of teacher's questionnaire Q4

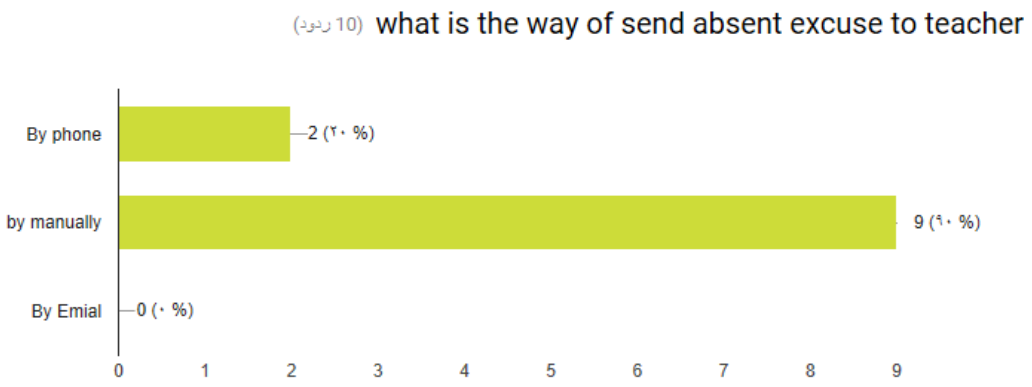


FIGURE C.16 Result of teacher's questionnaire Q5

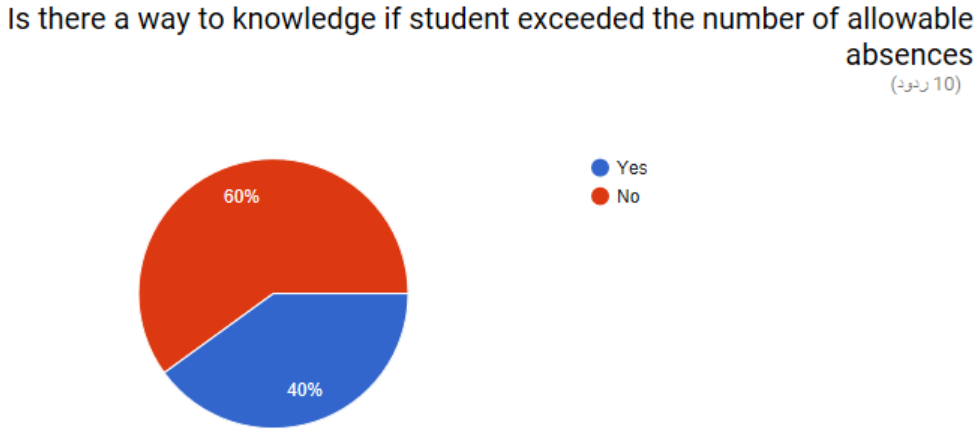


FIGURE C.17 Result of teacher's questionnaire Q6

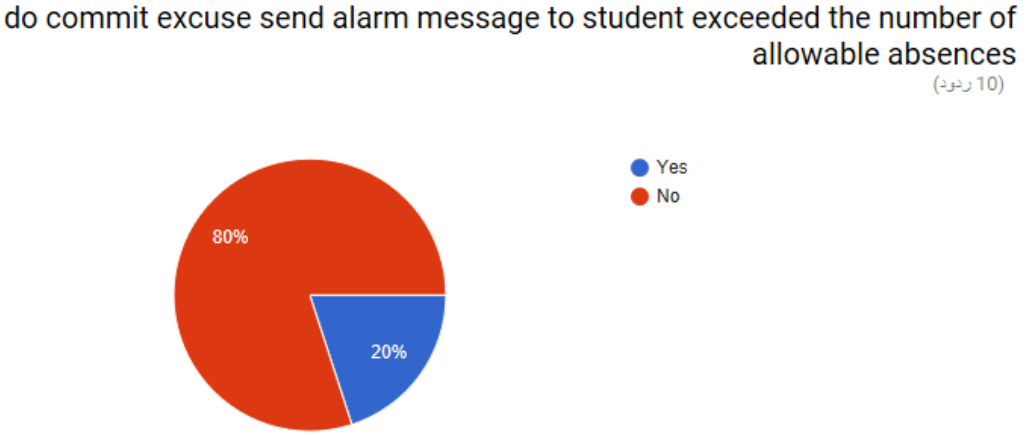


FIGURE C.18 Result of teacher's questionnaire Q7

(10 ردود) Do you see this application will save time of teacher and employee

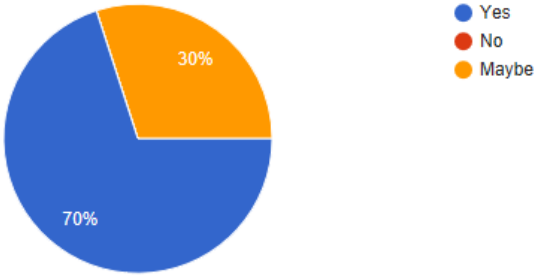


FIGURE C.19 Result of teacher’s questionnaire Q8