|  |
| --- |
| 1a |
| Faculty of Computers and Information Technology (FCIT) |
| Auto-attendes |
| Supervisor: Dr. Mohammad Naif Al-Atawi |
| Submitted in partial fulfilment of the requirements for the Degree of Bachelor of Information Technology |
| Academic Year: 2022/22 |

* **Abstract**

It is a mobile app for phones that facilitates many things that used to take a lot of time from the student and the Lecturer, and its main goal is to facilitate. Facilitate the attendance and the absent and other things such as knowing the locations of the halls and teachers for each college and lecture to facilitate the attendance in the fastest way to the hall or connected to the teachers’ offices‏

every student has the UniFinder app and every hall have an ibeacon that hold the lecture info, hall number lecturer that added by his student schedule, the beacon will sense the student who have UniFinder app and match his University ID to the Students of this current lecture, if the University 10 matches the student of this current lecture, then his (present) if not then send notification. This hall does not match your course

**Table of Contents**

**List of Figures**

**FIGURE**

Figer 1.Tragging . 5

Figer 1.TimeCamp . 5

Figer 1.Bonsai . 6

Figure 1. odoo 7

Figure 2. Waterfall method 8

Figure 3. Ibecon opreation 9

Figure 4. Class Diagrams 11

Figure 5. Sequence diagram 12

Figure 6. Activity diagram 13

* **Introduction**

For begaing let us review the the problems ,how much dose the lecturer loss time for attends, and some times he mistake for another Student, but our project idea can solve it, what about the student they mistake things that the project can solve such as what is the right hall , and is that my Curse, if the project has rich for the hight compltaion we can use it for things other then the unvirsty, such as Cafés, restaurants and touristic places, this project can have many ideas involvenig (area information, and sending it for the user).

* **Background to the project**

Given the importance of time in general for the student and the teacher in particular, and to take the maximum educational attainment

Our project focuses on saving time and facilitating many steps

* **Problem overview**

The college needs a way to facilitate the status of attendance and absence.

The students, the teachers, and visitors may have a problem to find the halls.

The teacher loses 5-10 min to take the attendance.

The teacher may conflict student's names.

* **Project Objectives**
* Speed of attendance and absence
* Find out where the halls are
* Knowing where the teachers are

**Aims and Objectives**

Therefore, we are working on an application for university students to facilitate the connection and save time for students to find halls and teachers

Auto-attendance and timer for the duration of the student's attendance of the lecture.

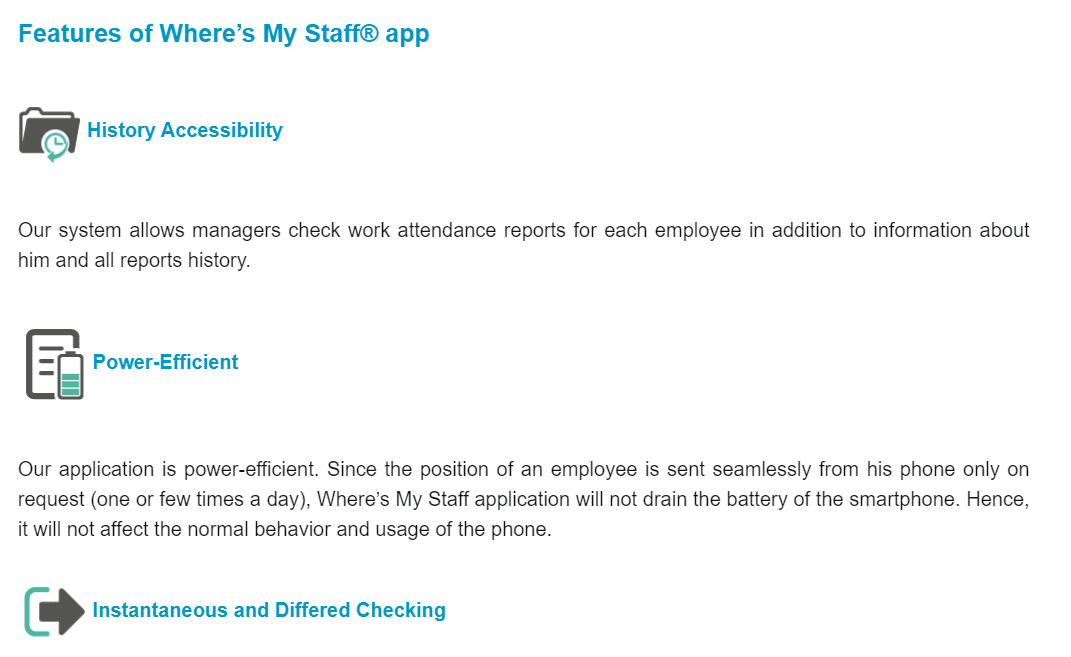
Our system will save the teacher and student time. Also, the conflict with students name will not be an issue.

* **Advantages**
* 1- Attendance and absence for all students automatically.
* 2- Temporary for the duration of the student's attendance of the lecture.
* 3- Hall status, is there a lecture or not.
* 4- Fetching the information of the current hall, for example (the lecturer, the Lecturer start timer)
* 5- Hall dates and locations.
* 6- Print the student attendance record for only the lecturer.
* **Drawbacks**
* The Cust of Ibecon
* Dose not specify the time if the Ibecon didn't indicate the student.
* If the student has his phone turned off then his absent.
* If the student has his phone turned off in the middle of class time, then in state time it will show less attends time.
* **Related works**

**tragging**

Where’s My Staff is a smartphone app developed by Tragging Company to help managers keep track of their employees during working hours. Office spaces and buildings

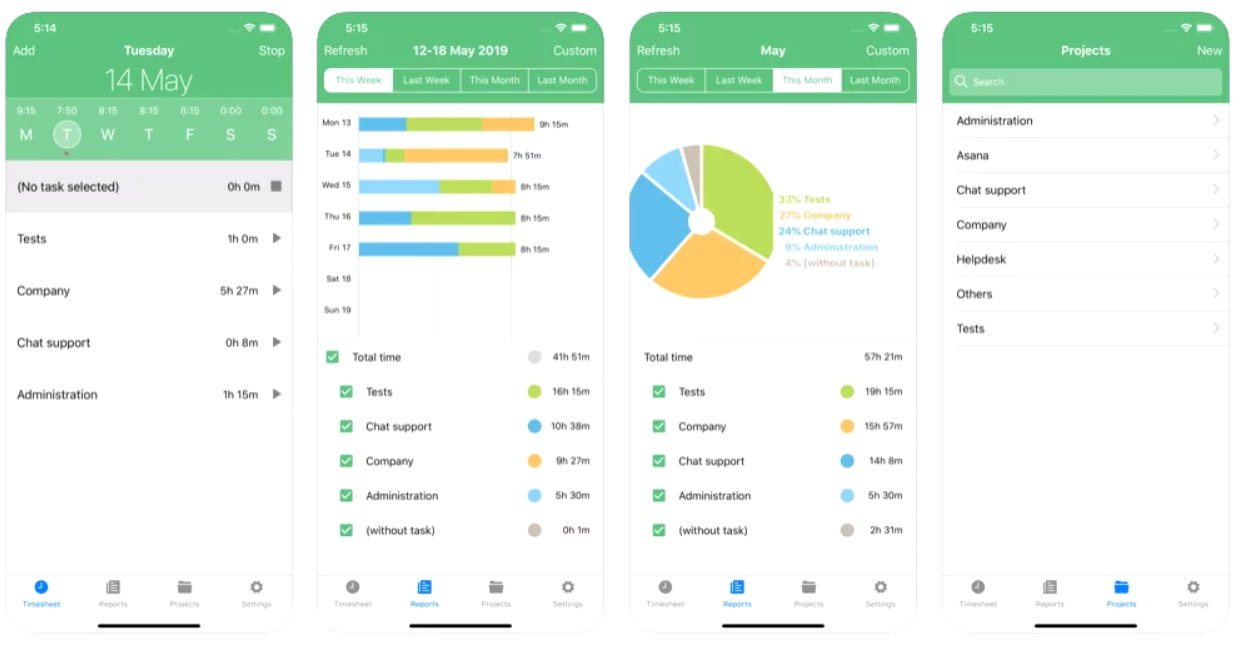
Our application is distinguished by the presence and absence feature



Figer 1.Tragging app.

**TimeCamp**

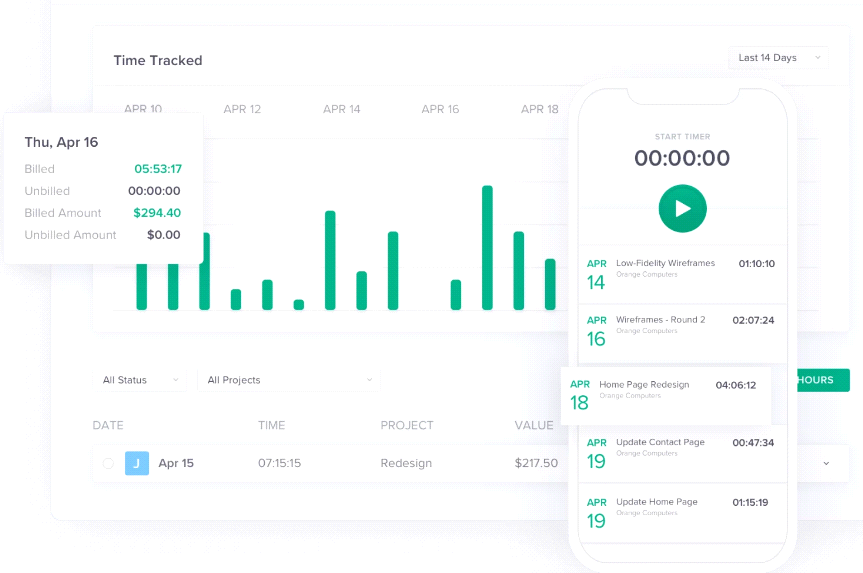
TimeCamp is a presence for employees and students, but it lacks many basic features, the most important of which are tracking, offices and places of presence



Figer 2.TimeCamp app.

**Bonsai**

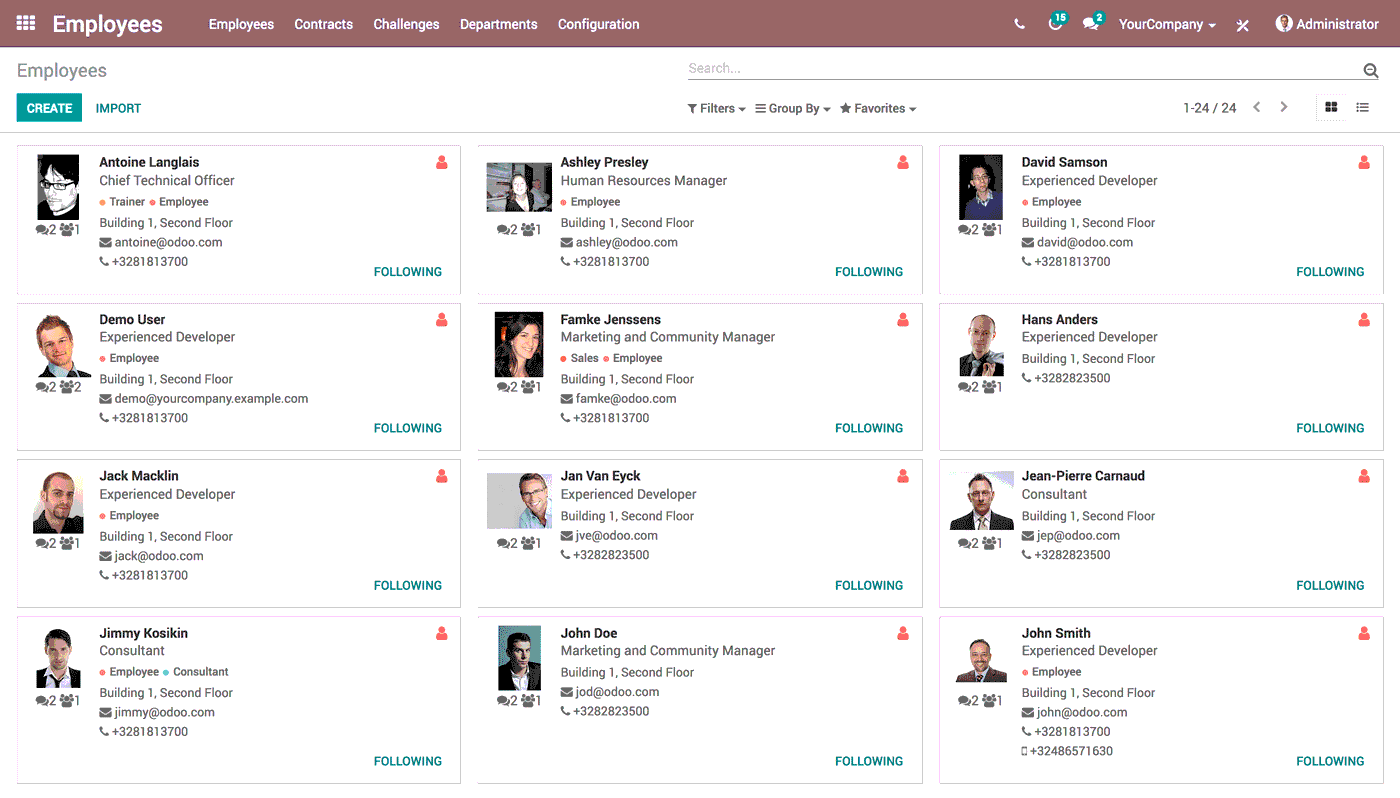
Bonsai is a set of tools for employees to manage attendance, and we are distinguished from it by the hall locations and making it available for use by students, not just employees and teachers



Figer 3.Bonsai wepsite.

* **Odoo**

This site has some points that are similar to our program, such as attendance management, but it lacks some features such as where the employee is located



Figer 4.odoo wepsite.

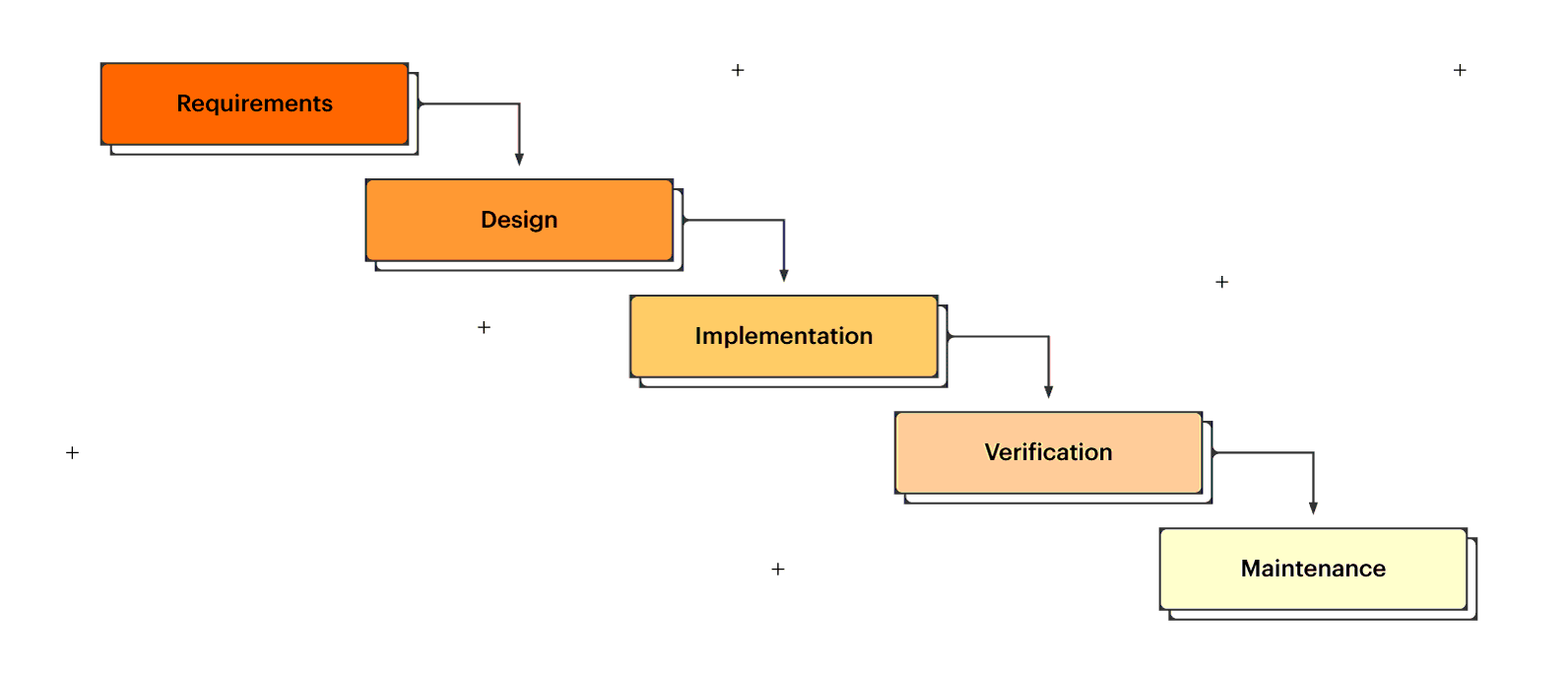
* **Menaitech**

This application is similar to our application, but it does not have options to adjust times and other options such as delay, and also there is no option for the location of the hall.

* **Methodology**

This project we well use waterfall methodology ,the waterfall method is project management approach that emphasizes a linear progression from beginning to end of a project. This methodology, often used by engineers, is front-loaded to rely on careful planning, detailed documentation, and consecutive execution.

the Waterfall process usually includes stages :



Figer.2 waterfall stages.

**Requirements :**

The Waterfall methodology depends on the belief that all project requirements can be gathered and understood upfront. The project manager does their best to get a detailed understanding of the project sponsor’s requirements. Written requirements, usually contained in a single document, are used to describe each stage of the project, including the costs, assumptions, risks, dependencies, success metrics, and timelines for completion.

**Design :**

Software developers design a technical solution to the problems set out by the product requirements, including scenarios, layouts, and data models. First, a higher-level or logical design is created that describes the purpose and scope of the project, the general traffic flow of each component, and the integration points. Once this is complete, it is transformed into a physical design using specific hardware and software technologies.

**Implementation :**

Once the design is complete, technical implementation starts. In this phase, programmers code applications based on project requirements and specifications, with some testing and implementation taking place as well.

**Testing :**

Testing needs to be done to ensure the product has no errors and all of the requirements have been completed, ensuring a good user experience with the software. The testing team will turn to the design documents, personas, and user case scenarios supplied by the product manager to create their test cases.

**Maintenance :**

Once the software has been deployed in the market or released to customers, the maintenance phase begins. As defects are found and change requests come in from users, a team will be assigned to take care of updates and release new versions of the software.

* **Requirements**

We will be able to derive the perfect understanding of requirements and analyses in this chapter and develop the software product. Stakeholder inputs and information gathered during the requirement gathering stage are used as inputs in this step. There are two types of requirements: functional and non-functional requirements

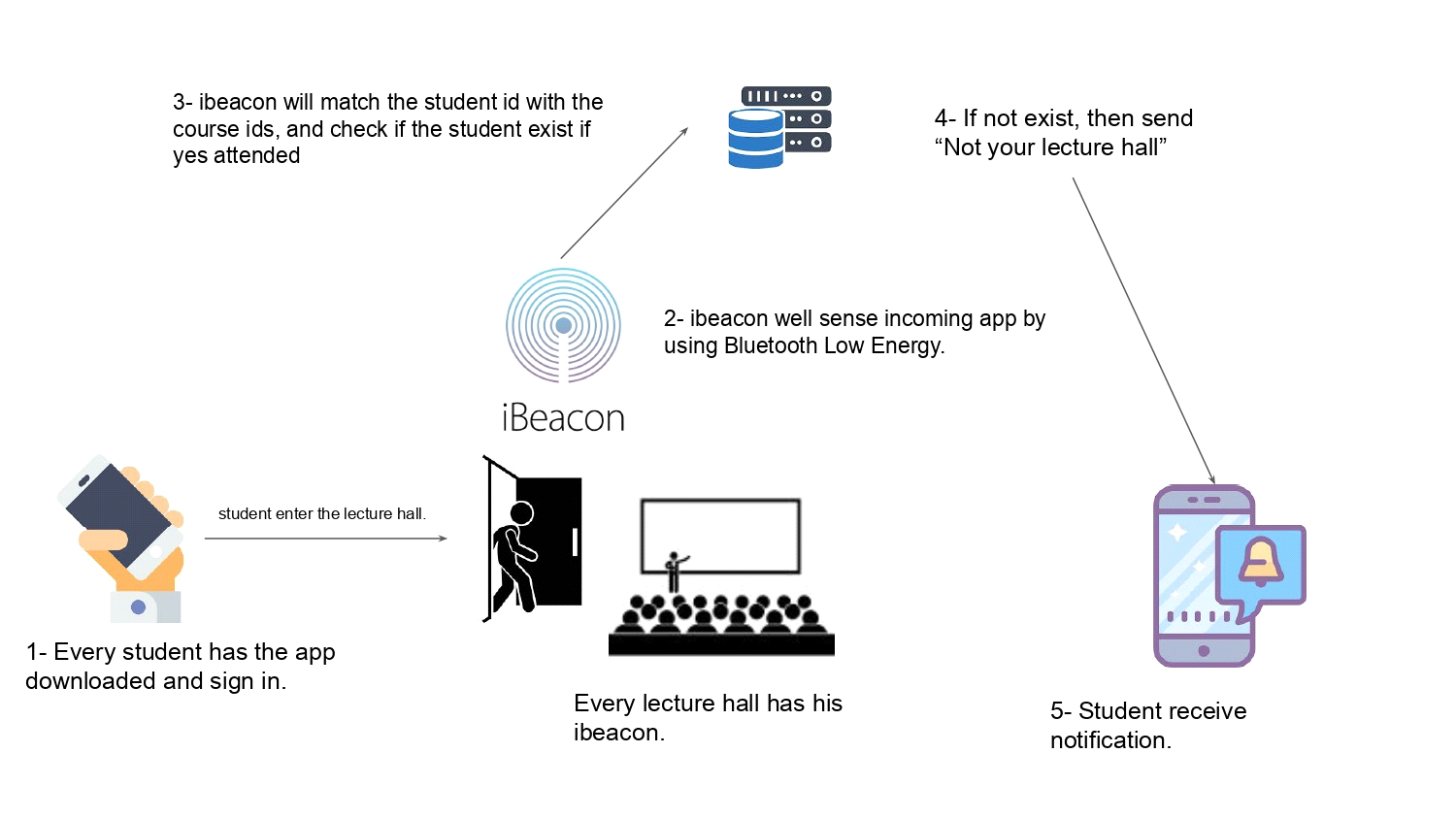


Figure 3. Ibecon opreation

* **Functional Requirements**

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks . This system has a many of functions that assist clinics with their processes and management,

For the softwear :

Administrator controls the accounts of lecturer and the student (Create, Modify, Delete).

Administrator controls the lecturer and (Create, Modify, Delete).

Administrator controls the accounts of the student (Create, Modify, Delete).

Lecturer requests the auto-attendants.

Lecturer requests print the state of class which is (students attendants time, students' attendance and absent).

Students request the hall name.

Students request the current hall state which is (subject of the class, Time of the class).

For the hardwear:

Buying ibeacon divase beacuse we can't bulid an ibeacon form the gruound zero.

Find ibecon that we can program not programed by the a company.

understand the student schedule to add it in the databace.

* **Non-Functional Requirements**

The non-functional requirement is a requirement that specifies criteria that can be using to judge the operation of a system, rather than specific behaviors [6]. The non-functional requirements that will provided in this project is usability, that is mean is a measure of how well a specific user in a specific context can use a product to achieve a defined goal effectively, efficiently and satisfactorily.

For the softwear :

Finding a ibeacon divec that allow us to program , we need to ask the company for permation.

Connecting the ibeacon to the system coding .

Connecting the ibeacon to the databace.

Bulding a server that hold all the students and lecturer.

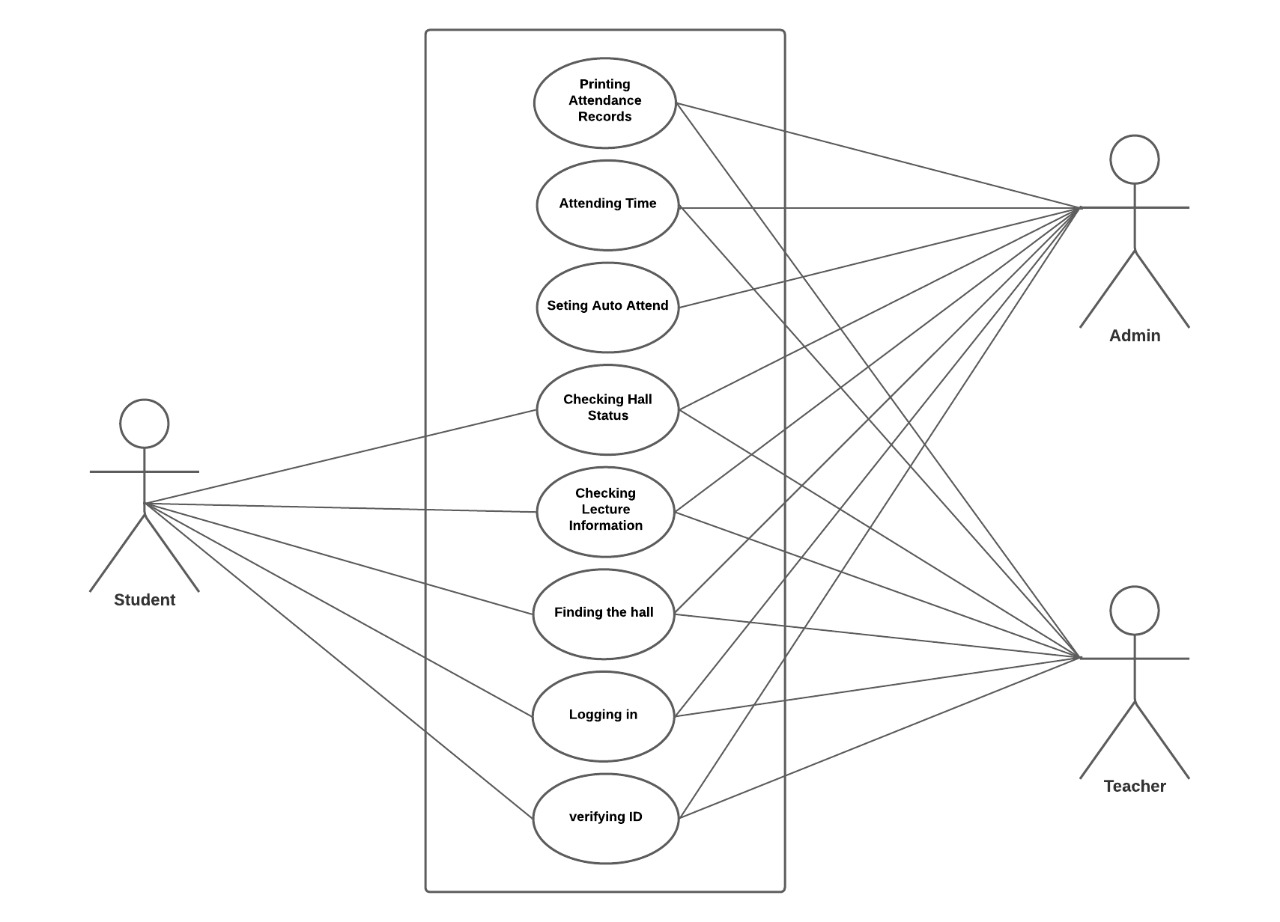
For the hardwear:

the cost of the ibeacon that we need every hall to have.

Ibeacon sencer range.

in case of hardwear errors we need to buy a new ibeacon.

dose the ibeacon stay on for timing the student attended.

* **Use Case Diagram**
* **Sequence Diagram**

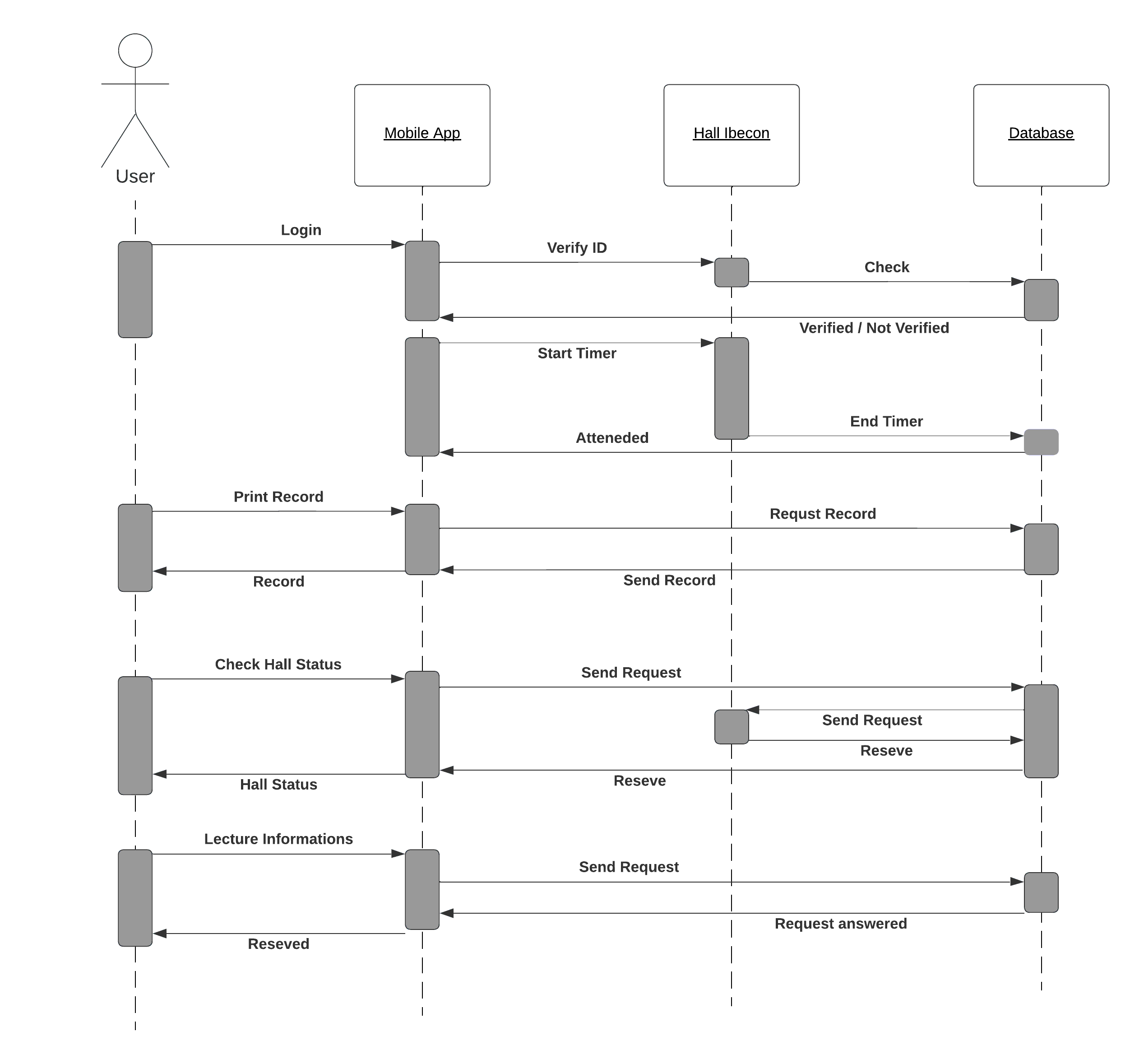


Figure 5. Sequence diagram

* **Activity Diagram**

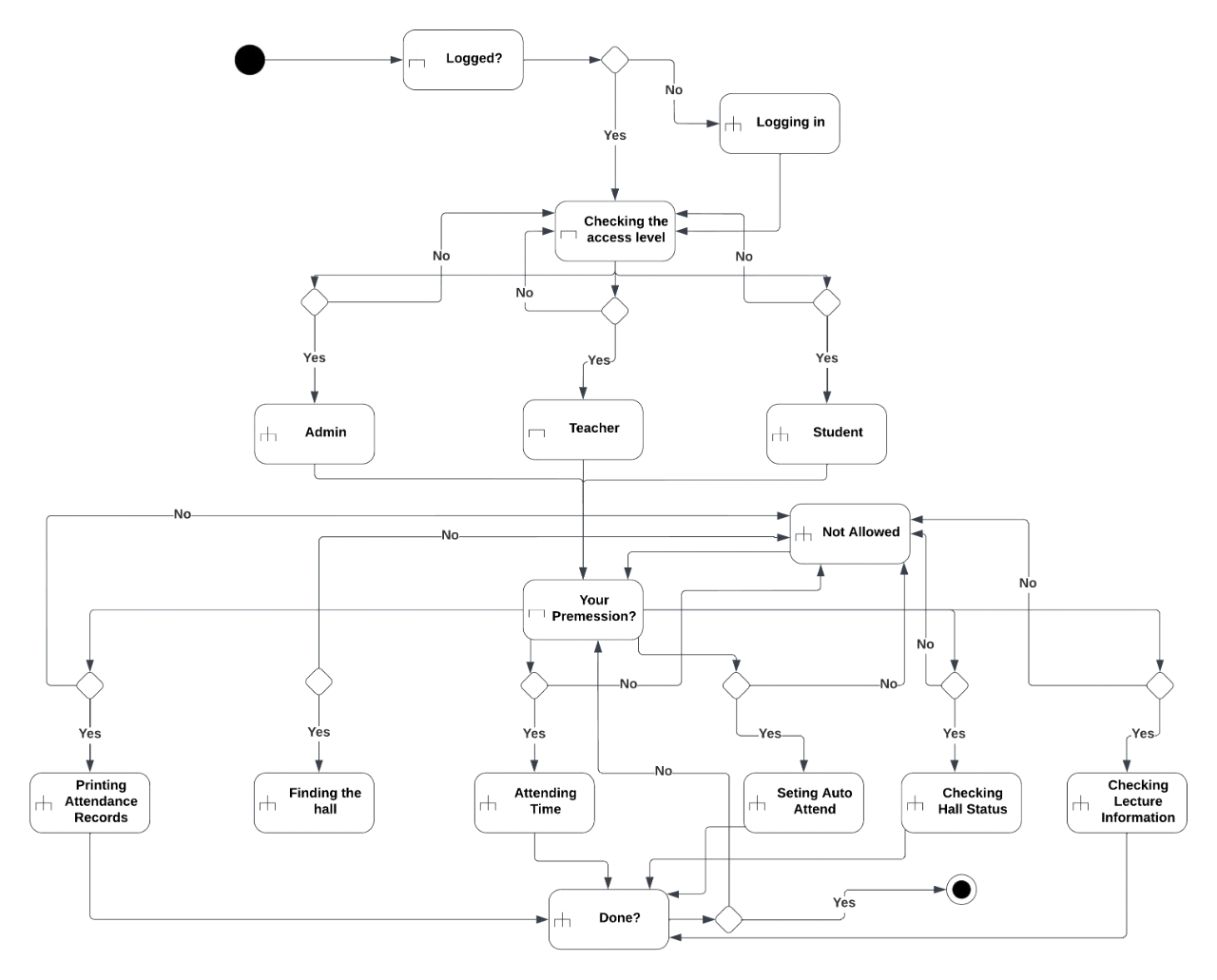


Figure 6. Activity diagram

* Data Design

Data dictionary is an inventory of data elements in a database or data model with detailed description of its format, relationships, meaning, source and usage [7]. The data dictionary for the project is shown in the tables below.

**Table 1. administrator**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key** |
| admin\_id | int(1) | PK |
| name | varchar(50) |  |
| username | varchar(25) |  |
| password | varchar(45) |  |
| email | varchar(45) |  |
| mobile | int(15) |  |

Table 2. Leuctrer

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key** |
| Leuctrer\_id | int(5) | PK |
| Leuctrer\_name | varchar(50) | FK |
| username | varchar(25) |  |
| password | varchar(45) |  |
| mobile | int(15) |  |

Table 3. Students

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key** |
| students\_id | int(9) | PK |
| name | varchar(50) |  |
| username | varchar(25) |  |
| password | varchar(45) |  |
| mobile | int(15) |  |

Table 4. Hall

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Key** |
| students\_id | int(9) | PK |
| Lucter\_name | varchar(50) |  |
| Luctrer\_name | varchar(25) | FK |
| Attendes\_time | varchar(45) |  |
| hall\_number | int(15) |  |