King Fahd University of Petroleum and Minerals
College of Computer Sciences and Engineering
Computer Engineering Department
COE 485 – Senior Design Project (T202)

Final Report

An Industrial Assets Tracking and Management System

Supervisor Name: ***Dr. Louai Al-Awami***		
Group Members		
Name	**ID**	**Signature**
Osama Bu Jwaied	**201661700**	
Yousef Almushayqih	**201565730**	
Khatim Alzahrani	**201418240**	

22-4-2021

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

According to a study by Hiscox, $50 billion is the annual loss for businesses caused by employee theft in the US only [1]. Not only this, but these costs are increasing annually by a percentage of 15% [2]. Asset theft causes financial and moral losses to organizations and companies. Financial losses result in the purchase of another asset that serves the same purpose as the stolen asset. In terms of morale, asset theft destroys the company's culture and destabilizes honesty among employees. Theft is not the only reason, but sometimes the loss of assets is due to moving them out of their default places without a permission.

Based on these problems, companies tend to build an asset tracking system to protect them from loss or theft. With the growth of companies' size and their inventory size, asset management and tracking software became an essential part of any company's cost-effective plans. Depending on an enterprise's size and operations, different types of assets can be available, ranging from consumable pens and pencils to expensive vehicles and industrial equipment.

1.1 The Problem

Managing assets is a challenging aspect of many enterprises and automating such an aspect could save a huge deal of money, effort, and headache. In addition, employee theft of assets causes huge financial damage to companies. The difficulty of tracking assets is directly proportional to the size of the company. The larger the company, the greater the number of assets there. Tracking assets becomes a very complex task. In addition, the importance of the assets varies based on their rarity and values. Tracking some type of assets is waste of time and money. In the other hand, there are assets that a company must track them.

Building an asset tracking and management system to help improve productivity and cost savings for companies is the project's objective. In addition, our service provides many companies an infrastructure to track assets. In this project, a website will be built that serves companies to track their assets. This website will be integrated with an infrastructure that will be provided to the company for getting suitable solution for tracking and managing their assets.

1.2 Possible Approaches

Manual approach

Manual tracking has more than one way. One of its forms is to employ a person whose primary task is to keep track of the location of assets. But one of the main problems with this approach is that the responsible employee cannot keep tracking all the assets at the same time. As a result, anyone can take any assets he wants in the absence of the responsible employee. Manual tracking may come in another way. For example, security cameras can be installed all over the place. But the problem with security cameras is the difficulty of tracking, and it also requires a responsible person to have access to the clips. Either way, the manual method is ineffective, and it is considered slow and costly to perform it.

Machine vision

The purpose of this technology is to install cameras that depend on computer vision technologies. Moreover, these cameras can analyze objects and identify people. This technique is more effective than manual approach. But this technology has some problems as it consumes more energy. One of the problems is that this technology can only recognize objects and is not able to differentiate between assets that have the same body. For example, if there are two coffee machines having the same shape, the camera will identify both and consider them the same. What if an employee came with his own coffee machine to the company? The camera will identify it as one of the company's asset.

Identification technology

Automatic identification technology is considered one of the best technologies for tracking assets. This technology is based on automatic identification of objects. It has several types, the most famous one is RFID technology. This technology is based on identifying objects by using of radio waves. It does not require Line of Sight to identify objects. This technology consists of two main components, the reader, and the tag.

RFID technology is considered the best way to track assets for many reasons. First, the objects are automatically identified just when the reader reads the tag. This

reduces human interaction and increases automation. Second, many tags can be read simultaneously.

Using this technology, assets can be traced by connecting the readers to a computer device that sends the information to the cloud if an asset is read. The computer sends both the ID of the reader and the tag ID. Using these IDs will allow the system to display current location of the asset.

Table 1. Summary of possible approaches. Selected approache is hilighted.

Approach	Description	Pros	Cons
Manual approach	Tracking assets by using human that register when and where the assets available, and with who.	• Less electronic resources • Accuracy • Environmental	• Expensive
Machine vison	Technological method provides automatic imaging-based inspection and analysis for tracking the assets	• Accuracy • Reduce Cost	• Specialists needed. • Complexity
Identification technology	Tracking assets by scanning tags using RFID, GPS, or BLE.	• Fast • No human interaction • Economical	• More magnetic field radiation • More electricity consumption

In our project we will focus on tracking stationary assets of organizations by using **RFID technology**.

- **Requirement**: registering assets movement around building**.**
- **Explaining:** this automatic tracking system will work with assets that constantly moving inside the building from place to place. The assets will contain a tag card, and each partition of the building going to contain RFID to trace the asset. In

addition, the assets will be linked with the system, which contains a database to know the location of the assets.

- **Pros:** Cheap – Indoor localization.
- **Cons:** Cannot know who use the asset – No real time tracking.
- **Work with:** Furniture – Chair – Table PC - Book – Coffee Machines – Projector – Printers – Remote control.

1.3 Potential Impacts

Potential Benefits:

Advanced asset tracking system will help the companies to reduce the cost and time of losing the assets and buying now ones. So, enterprises overhead cost will be decreased which will lower the cost for consumers and improve their economy.

Furthermore, the system will help the companies to trace their assets movement inside the company, and to locate last location the asset available on. So, they can pick it easily from that location instead of searching the whole company.

In addition, the system will help making specific location that assets can be used on, and the system will help detecting if some assets get out of its allowed range in the company.

- Accuracy.
- Profits.
- Faster and cheaper movement of assets.
- Assets sharing.

Potential Pitfalls:

The most import disadvantage of RFID tracing system is the magnetic field radiation which has a negative effect to people health. Passive RFID reader technology consume electricity and transform it to radiofrequency that used to read the passive tags ID number. this radiofrequency has a great affect to people health and welfare by infected them with magnetic field radiation. This radiation can cause headaches, anxiety, nausea, fatigue, and loss of libido to human.

- More electricity consumption.
- Increase magnetic field radiation.

- Affect environment.
- Affect human health.

Table 2:Table . Summary of potential impact of the selected approach. Potential negative impacts are highlighted.

Impact	Description
Economic	**Local:** reduce overall tracking cost for the company**.** **Global (if it applies):**
Health	Little impact for health because of electromagnetic radiation.
Environmental	Ignorable impact for Environmental because of electromagnetic radiation. Our system will help reducing the amount of waste by reducing the number of assets loss.
Societal (people's welfare & safety)	It will improve overall human quality of live by reducing time waste and working effort.

2. REQUIREMENTS, SPECIFICATIONS, AND CONSTRAINT

Requirements:

1. The system shall be lower cost than the manual system.
2. The system shall provide a user-friendly interface to perform different management operations such as when, where, and by whom the assets were used.
3. The system shall be automated, with minimum human intervention.
4. The system shall be able to approximately track and identify assets within the company's building.
5. The system shall be secure and provides proper measures against tampering and hacking.
6. The system shall notify the admin when an asset moves from its default place.
7. The system shall provide complete information about assets (Name, Type, location, ID).

Specifications:

1- Every administrator shall be able to add & delete assets.
2- An alarm must be triggered when an asset moves without permission.
3- Every employee and administrator shall be able to see details of assets (Name, Type, location, ID).
There are two types of memberships for every subscription (organization) in our service. The first type is managers, and it has the powers to add and delete assets, while the second type is employees. They can only track the assets and obtain their information.
4- The system must allow users to borrow assets when needed.
5- The system must send an email when an employee borrowed asset and he did not return it after the deadline.
6- The System shall be able to show last approximated location for the assets.
7- RFID reader shall send data to the cloud when assets move out of their ranges.

8- (3) Long range RFID readers will be used to track assets far away assets. Passive RFID tags will be used because there is no need to changing or charging the battery.

9- (4) RFID readers will be distributed all offer the company to track assets movement wherever it goes and send it location to system.

10- (5) Apply security protocol such as HTTPS, SSL, and SSH to protect the data, and provide security protection against different types of attacks such as cross site scripting (XSS), cross site request forgery (CSRF), and SQL injection attacks.

11- (6) Well-designed website application will be available to company administrator and employees to show assets status

Constraint:

1. Duo to subscriber need the total cost of tracking the assets should be lower than manual work. Employee that works in managing the assets will lost their job, which will lead to bring hate to the company.
2. Duo to some need of users. The user interface shall be simple and easy to use.
3. Duo to subscriber and employees need the system shall provide much less human interaction.
4. Duo to selected approach system shall be able to capture assets movement everywhere in the company and send assets information and status.
5. Duo to some regulations the system shall provide proper security protection.
6. Duo to subscriber and employees need the system shall provide complete assets information and status.

Table 3: Project requirements, and the corresponding specifications and constraints.

Requirement	Corresponding specification(s)	Corresponding constraint(s)	Justification
(1) Lower cost	Less number of employees needed to track the assets. Lower assets lose.	Lower total cost of tracking assets than manual work. Hate from employee who lost their job.	Only way to reduce cost is to automate tracking system and decrease the number of employee.
(2) User-friendly interface	Making a practical well designed website application.	User interface shall be simple and easy to use.	
(3) Minimum human interactions	Using long range RFID readers & passive UHF RFID tags.	Duo to subscriber and employees need the system shall provide much less human interaction.	Long range RFID readers & passive UHF RFID tags will be used to maximize the automation and reduce human interaction as much as possible.
(4) Provide assets location.	Distributed RFID readers will track asset's tag and send it location to the system.	Capturing assets movement and send it information and status wherever it goes.	

(5) Security	Apply security protocol such as HTTPS, SSL, and SSH protocol. Provide security protection against different types of attacks.	Duo to regulations the system shall provide proper security protection.	Providing a service over the internet will make the system vulnerable to different types of attacks from hackers.
(6) Provide assets information	Website application will be used to show assets information	Duo to subscriber and employees need the system shall provide complete assets information and status.	

3. WORK PLAN

Table 4:Working plan.

Student Week	Osama Bujwaied 201661700	Yousef Almushayqih 201565730	Khatem Alzahrani 201418240
1	Reading about assets tracking techniques, implementation, requirement. Scheduling and organizing the meeting working times between group member		
2	Writing Action Plan. Distribute the work between group members		
3	Gather more information about tracing assets by embedded devices technique. Learning about cloud working mechanism	Gather more information about tracing assets by borrowing management technique.	Gather more information about tracing assets by registering movement around building technique.
4		Choosing the framework (Django)	Choosing required component for assets tracking
5	Developing use cases, activities diagrams, and writing the mid-term report		
6 Implementing phase	Building a simple database	Build a basic Django app	* Examine required hardware cost to implement the project. * Simulate RFID using Packet Tracer
7	Build a database using MySQL and connect it to AWS	Linking static files needed to Django	
8	Start building the dashboard	Building the main page for web application. (Welcome Page)	Learning **Django** high-level Python Web framework
9	Build a database inside Django SQLite	*Deploy Django using AWS cloud.	*Connect the packet tracer server to outside world.

		*Participating in developing the logic for automated assets borrowing	*Participating in developing the logic for automated assets borrowing
10	Building Django web pages and connect them to database (Dashboard). **Such as** -Admin page - Subscriber page - Employee page - Tags pages	*Enhancing the frontend. *Design login and logout pages. *Adding form to the dashboard so that an employee can submit a request for assets borrowing + Add asset	* Developing a script for sending both asset's tag and employee ID after reading from RFID reader.
11	Upload Django website to GitHub and deploy Django website into Heroku (PaaS)	*Add restriction for login to web pages using admin, subscriber, and employee account. *Linking Django to Gmail to send an email if an employee did not return the asset. * Writing a script to check borrowing tables & send an email if needed.	Simulate Advanced coded RFID tracking system using Packet Tracer and connect it to Django framework.
12	Revise the whole system and modify it. Improve Django website Design.		
13	Test the demo	Test the demo	Test the demo
14	Making final report & presentation		

Note: Details about work distribution will be provided in the appendix on Microsoft Team

4. SYSTEM DESIGN

4.1 System's Behavior:

Use-Cases

The use-case diagram shown in below illustrate the use cases to each actor.

We have five use cases.

1- **Login**: since we have two main types which are employees and administrators, the dashboard and the authority will differ depending on the type.

2- **Get Tracking information**: This use case is one of the essential parts of our system. From this use case, the admin will know the location of the asset if it is in the building.

3- **Borrowing**: Our service provides the ability to an employee to borrow assets. Even the employee needs to take that asset outside the building.

4- **Send email**: an email will be sent if the employee did not return the asset he borrowed after the deadline.

5- **Add & delete assets**: the administrator can add and delete asset when needed.

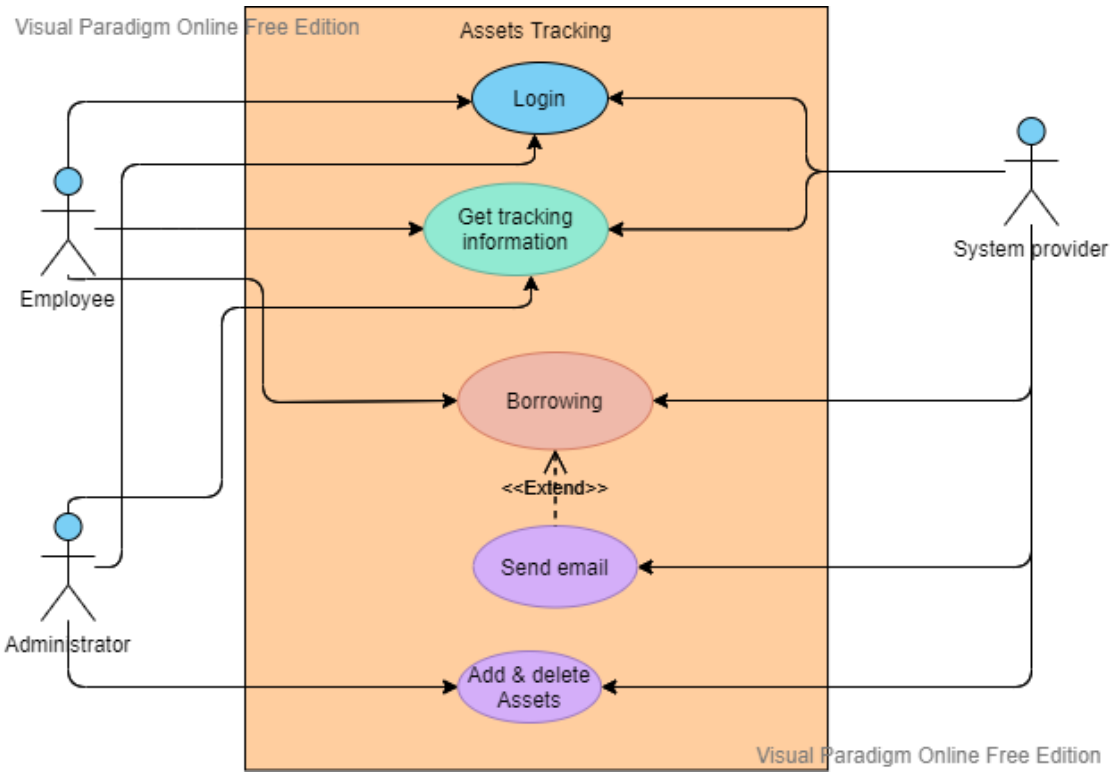


Figure 1:Use Case Diagram

Activity Diagrams

Table 5:Activity Diagram for Login User

Use Case	Login User
Actors	Employee, admin
Description	Login to the dashboard
Preconditions	None
Postconditions	None
Assumptions	None

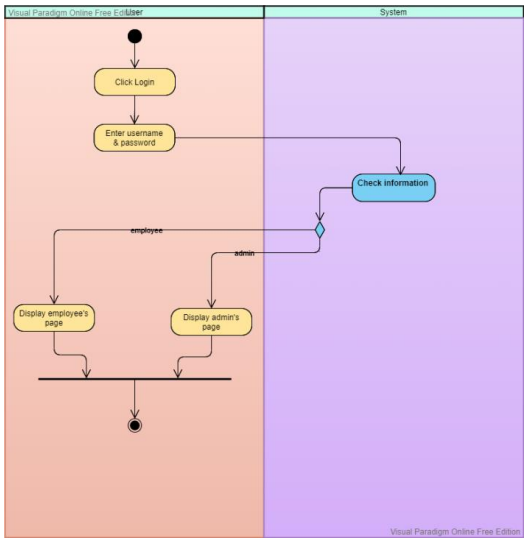


Figure 2:Activity Diagram for Login User

Table 6:Activity Diagram for Get Tracking Information

Use Case	Get Tracking Information
Actors	Administrator
Description	Tracking assets movement and current location by the system.
Preconditions	Login, Add assets
Postconditions	None
Assumptions	

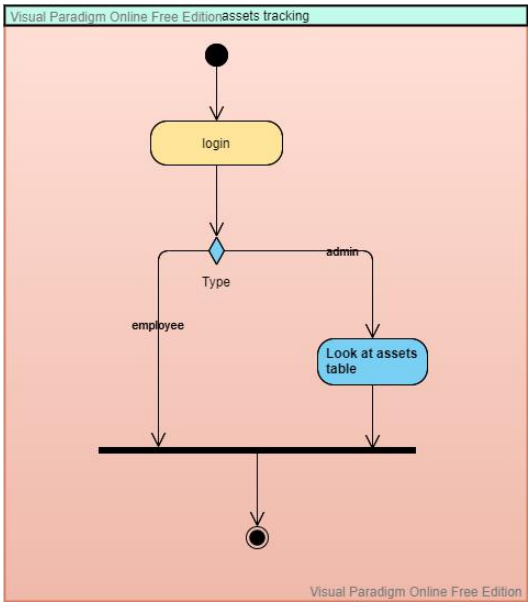


Figure 3:Activity Diagram for Get Tracking Information

Table 7:Activity Diagram for Borrowing an asset

Use Case	Borrowing an asset
Actors	User
Description	User process to borrow the asset so he can use it out of company region
Preconditions	login, add an asset.
Postconditions	None
Assumptions	

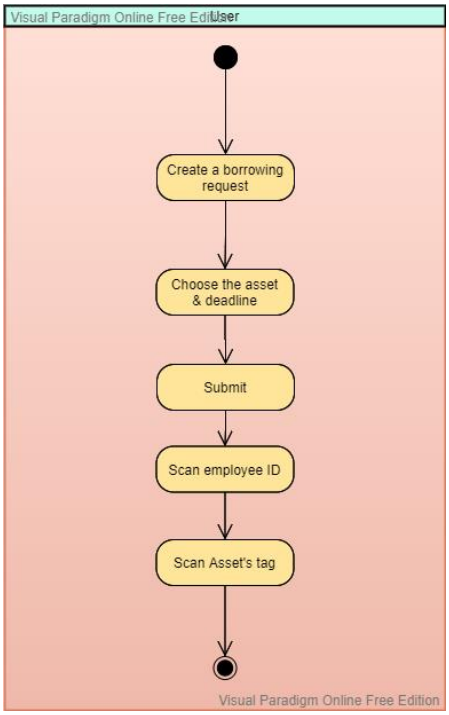


Figure 4:Activity Diagram for Borrowing Asset

Table 8:Activity Diagram for Deleting assets

Use Case	Deleting Assets
Actors	Admin
Description	This diagram illustrates how an administrator could delete an asset
Preconditions	login
Postconditions	None
Assumptions	None

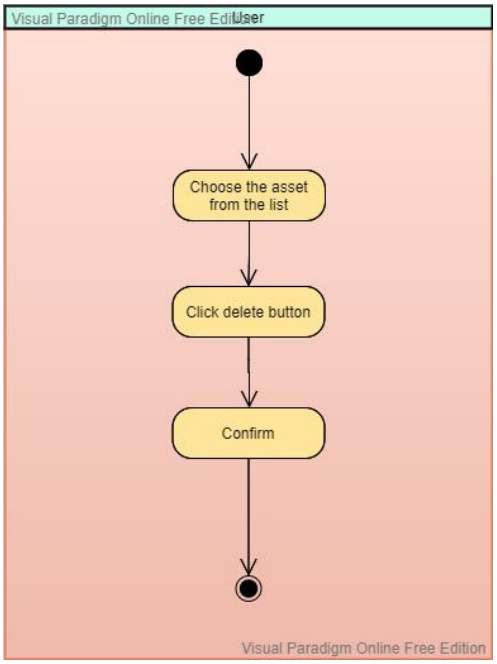


Figure 5:Activity Diagram for Deleting Assets

Table 9:Activity Diagram for Adding Assets

Use Case	Adding Assets
Actors	Administrator, Subscriber
Description	The mechanism used to add the assets from the system using the tags ID that were assigned to them.
Preconditions	login
Postconditions	None
Assumptions	None

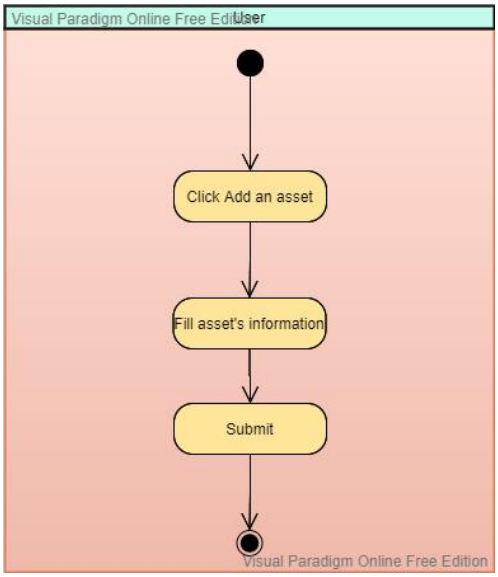


Figure 6:Activity Diagram for Adding Assets

Table 10:Activity Diagram for Send an email

Use Case	Send an email
Actors	Assets
Description	If the employee did not return the asset after passing the deadline, an email will be sent to the admin to let him check the list. SBC will be programmed to do HTTP request to a dedicated URL for every 24 hours. Every time a request happened; the function will compare.
Preconditions	Login, borrowing.
Postconditions	None
Assumptions	None

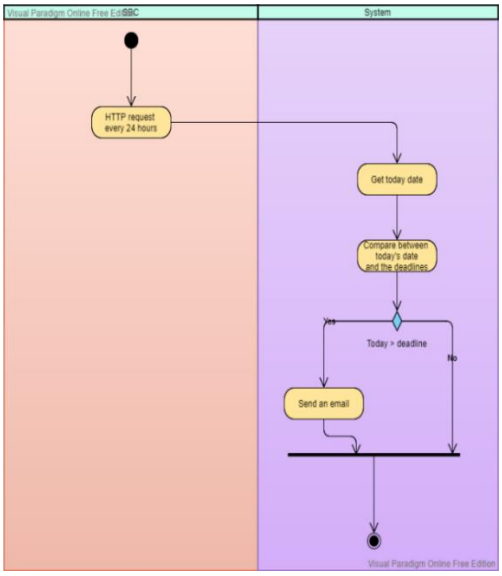


Figure 7:Activity Diagram for Sending Emails

4.2 Product Concept

All companies contain assets within the company building. Some examples of these assets are coffee machines, monitors, desktop computers, and so many more. Often, these assets are borrowed or moved by employees for some reasons. For example, an employee might want to move a monitor from the office to a conference room for a purpose. It often requires significant human effort to monitor and control these assets to protect them from theft or loss.

One of the problems is when an employee wants a certain asset but when he goes to its usual place, he does not find it there. Often employees tend to search for assets by asking other employees. Another problem is when an asset is stolen, it costs the company financial losses.

The main goal of the project

Our service comes to provide an infrastructure and a website providing a friendly user interface to facilitate and automate the process of tracking assets within the company using RFID technology.

Tracking

Our service provides companies with RFID readers, passive tags for both employees and assets, and Siren. Then, we configure RFID readers in distributed locations inside the company to track assets. by doing this, indoor assets tracking will be achieved. When an RFID device read a tag of an asset, the computer associated with the RFID reader will send the tag ID and the reader ID to the cloud and then store the new location in the database. The new location will be determined based on the reader ID. Assets' locations will be displayed in the dashboard for employees.

Assume there are two readers, reader 1 which is in the office 1, and reader 2 which is in office 2. If an employee moved the asset from office 1 to 2, reader 2 will identify the asset then will send both Asset's tag and reader id to the cloud, based on reader id, the cloud will change the update the location. See figure X.

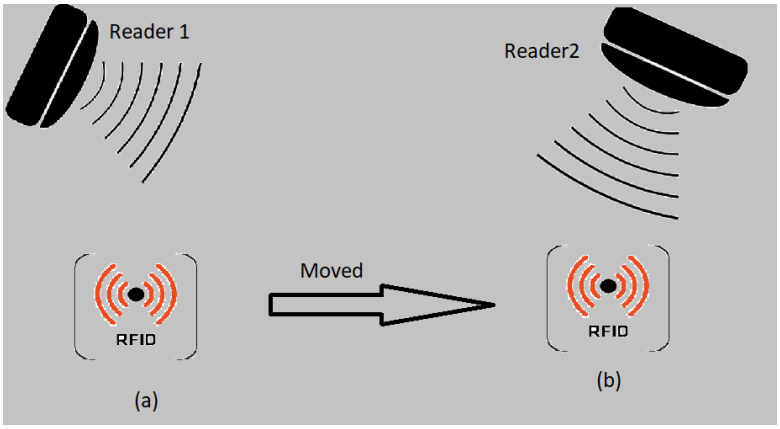


Figure 8: An asset moves from place to place.

Authorizations

The website benefits all company employees, but the powers differ. Authorizations have been divided into two types:

- **The employee**, who can only track asset and request a borrowing for an asset.
- **The administrator** who can track assets, in addition to the possibility of adding or deleting an asset.

Self-borrowing service

Our service also allows the possibility of borrowing the asset if an employee wishes to borrow. When the employee wants to borrow some of the assets, the employee should go to his dashboard, then he should submit a request for borrowing. After submitting a request for borrowing, the employee should scan his ID on the tag reader and then scan the tag of the asset. After doing this, the employee can take the asset out of the company. If the employee failed to do these steps and he took the asset, the alarm siren will be triggered. We have developed a flowchart to describe the system of borrowing. (Figure x). Based on the borrowing feature, there are two readers in each room, a reader for assets tracking, a reader for both asset's tag and employee ID.

Scenarios

The first scenario: It is when a person takes an asset without asking to borrow it from the website. In this case, after taking the asset out of its usual place, the reader reads the tag. Then, the asset's tag and the reader ID are sent to the cloud. Then the cloud uses the asset's tag and searches the table. If the asset's tag is not found, the cloud will send a response to the client that no enrollment was found in the borrowing table in the database. An alarm in the admin's office will be set and an LCD will print the tag's ID and the location of the reader that last read the tag.

The second scenario: It is when a person requests to borrow a specific asset. This request will be added to the borrowing list. To complete the process, the employee should scan his ID and the tag ID. If we assume that he did not scan and took the asset out of the room, the reader will read the tag and send both the asset's tag and the reader ID to the cloud. After the search process, the asset's tag will be found inside the borrowing table, but the values that check the scanning are zero, and this means that the employee did not scan his ID nor the asset's tag. Therefore, the cloud will send a response to the client, and the alarm will be triggered.

The third scenario: It is when a person requests to borrow a specific asset from the website. After submitting a request, this request contains the employee ID, the tag ID, and the deadline. there are two values that will be associated with each request in the backend. These values will be set to 0. The first value is called ***"EMPLOYEE ID SCANNED"*** where the second is ***"ASSET ID SCANNED"***. The employee will scan his ID. Then, the value of "***EMPLOYEE ID SCANNED"*** will be set to 1. After this, the employee also will scan the asset's tag and the value of ***"ASSET ID SCANNED"***. will be set to 1. Now, the employee can safely take the asset out of the room. If the reader read the tag, the tag will be sent to the cloud. The cloud will search the table of borrowing and it will find the asset's tag. Then, the cloud will check the values of both "***EMPLOYEE ID SCANNED"*** *and* ***"ASSET ID SCANNED"***. Since both are one, no siren will be triggered.

If the deadline passed and the employee did not return the asset, an email will be sent from the cloud to the administrator.

All scenarios have been demonstrated in figure 8.

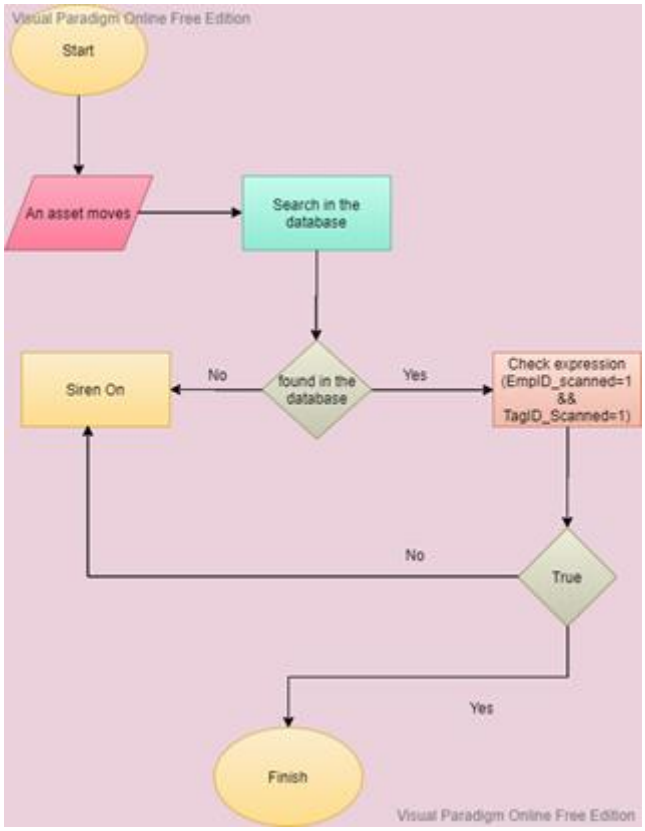


Figure 9:Product Concept Scenarios

4.3 System Specification

- **Bandwidth**:
  - 850 - 950 MHz for RFID reader and passive tags.
  - The system will need stable internet connection that is 10MB/s at minimum.
- **Latency**:
  - 10 – 100 ms for RFID reader and passive tags.
  - The data should be updated in the server in each minute.
- **Quality**:
  - **Accuracy:** 10 – 50 m^2 approximated assets location.
  - **Coverage distance:** 3 - 10 m per RFID reader.
- **Security:**
  - **Heroku:** provide security to network, data, and system. Network security can be achieved by Heroku using firewall, DDoS mitigation, spoofing and sniffing protections, and port scanning. Also, Heroku apply system security by application isolation, system configuration, and system Authentication ("Heroku Security", 2021)**.**
  - **Django:** provide protections against cross site scripting (XSS), cross site request forgery (CSRF), and SQL injection attacks. Also, Django allow the programmer to deploy his website using SSL and HHTPS protocol ("Security in Django", 2019).
- **Form factor:**
- **Power consumption:**
  - **Tags:** 0 power consumption from tag because we will use passive RFID tags which do not consume electricity and do not need battery.
  - **RFID Reader:** nearly 100w per each reader**.**
  - **Company power consumption:** number of RFID reader * 100w. It the end the cost of electricity is so low and ignorable comparing to the benefits provided by the system.
- **Human-machine interfaces:**
  - Employees and Administrators can use their own personal devices such as phone and computer to interact with system using the website that wo going to provide it to them.

Table 11: Different factors considered in the design.

Design Factor	Consideration
Economic Impact	The system will be mostly automated to reduce the cost of hand work employee as much as possible and increase the profit for the facility. Therefore we will use passive RFID tags that do not to be recharged, and long RFID reader that can capture the tags from fairly far distance. Also, these tags and RFID reader has an affordable cost. In addition, the system will reduce the amount of wasted assets, and save the cost of buying new one. This can achieve by providing the website that can provide to facility workers the current location of the assets.
Health Impact	Reducing electromagnetic radiations that going from RFID readers that can cause health problem. Because of that we used passive RFID tags and efficient RFID reader that do not consume a lot of electricity and produce a low number of electromagnetic radiations.
Environmental Impact	Our system will minimize the amount of waste. This can happen by tracking the lost assets and then reuse instead of buying other one.
Societal Impact (people's welfare & safety)	The system will be mostly automated to improve company facility quality of live. Therefore, we will use passive RFID tags that do not to be recharged, and long RFID reader that can capture the tags from fairly far distance. So, facility workers do not bother their self-doing a lot of work.

4.4 System Architecture and Deployment

Service Provider side:

The web server is deployed on the Heroku cloud. The website was built using Django framework. This framework provides models for the backend, whereas the frontend is done using HTML and CSS. In addition, the database Sqlit3 that we used for this system is integrated within the Django.

Subscriber side (company):

On the other hand, the company side contains some hardware components such as RFID readers, RFID tags, Single Board Computer (SBC), a siren and LCDs. In each room where there are many assets stored, two reader will be placed in there, one at the doorstep to prevent anyone from taking any asset without permission, and the other reader will be to scan employee and asset IDs in case of a borrowing request. In addition, an LCD will place besides the latter reader to print meaningful messages to the person scanning the ID of either an asset or an employee. All these readers are connected to the SBC which is the part that is responsible for communication between the cloud and the RFID readers.

The use of the siren is to alarm the admin of an unauthorized movement of an asset. That asset's ID and the location of reader where the asset was read will be printed on the other LCD. Figure 9 shows the deployment diagram for this system.

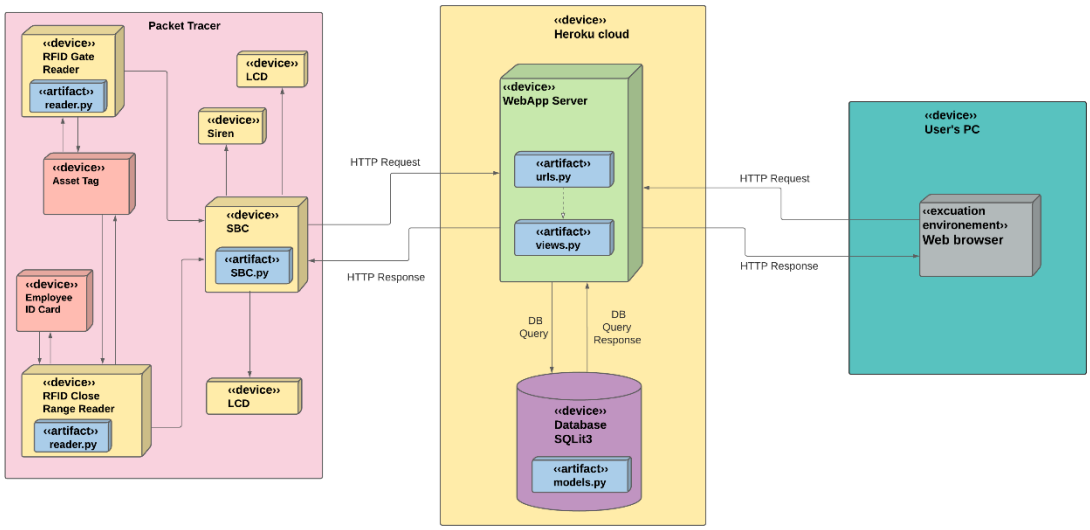


Figure 10: Deployment Diagram

4.5 Compliance with Standards

Table 12:Standards used throughout the project design.

Standard	Purpose	Where & why it is used
EPC UHF Gen2v2	communication protocol	It is used on the client side to identify the tags.
HTTP	Communication protocol (application layer)	this is the protocol that will be used between computer's client and the cloud. When a reader detects a tag moving a way, the RFID reader should send GET request to the cloud. In addition, this protocol will be used between employees' devices (Phones, laptopts) and the cloud when accessing the dashboard
HTML	language for structure web pages	It will be used when developed the dashboard and the main page.
SQL	A language for managing data	For storing data such as assets' location, tag IDs, borrowing table, usernames, names of employees etc.

5. COMPONENT'S DESIGN: -

5.1 Packet Tracer Simulation.

The client-side of this system consists of several devices. These include RFID readers, RFID tags for both assets and employee's identification cards, and a Single Board Computer (SBC) that will connect the readers and the cloud server. Due to the nature of the current situation, this part of the system will be emulated using Cisco Packet Tracer. A description of how each component was designed within Packet Tracer and what their real-life equivalent would have been.

5.1.1 RFID Readers

Specifications and constraints

The RFID reader's job is to detect the movement and identify assets when they pass by it and send the data to be further processed. This will help achieve the two essential functions of the system, which are tracking and identification.

Table 13:Specifications of the UFH gate readers and HF close range readers

Reader Type	Specifications
UHF Gate RFID Reader	- Protocol ISO18000-6B, EPC Class 1 Gen2 (ISO18000-6C) -Reading Rate 80 – 100 tags/second -Mounting Wall Mountable Protection Grade IP 40 -Work Frequency Standard ISM 865-868MHz, 902~928MHz (frequency can be customized) -Frequency Hopping FHSS -Power Output 0-30dBm (adjustable) -Antenna One – Four Antennae, TNC Connector Interface Rs232, RS485, Wiegand 26/34 (TCP/IP, Wifi can be customized) -Operating Mode Answer, Active, Trigger -Read Range Upto 8 meters with 8dBi Antennae and up to 15 meters with a 12 dBi Antennae -Write Range 0-5 meters

	-Reading Clue Buzzer and Led -Input Power Supply External 100-240 V AC Adaptor -Dimension 225 ×170 ×40mm Reading Rate ---Software Programmable -Regulatory Compliant with CE, FCC
Close Range HF RFID Reader	-Operation Baud Rate. 9,600 – 115,200 bps -Supply Voltage . Regulated 5 V DC -Supply Current 200 mA -Speed. 12 Mbps (High Speed) -Standard. ISO 14443 A and B Protocol. Mifare® Classic protocols, Mifare® DESFire protocols, Mifare Mini protocols -Smart Card Read/Write Speed . 106 kbps -Dimensions. 120.5 mm (L) x 72.0 mm (W) x 20.4 mm (H) -Reading distance of up to 50mm -Certification/Compliance CE, FCC, RoHS

Design and Verification

There are two scenarios to read the RFID tags in this system, and two types of RFID reader technology are needed. The first scenario is to detect if an asset was taken and moved from its place without permission. An RFID gate reader is suitable for this purpose. The second scenario is scanning an asset whenever someone request to borrow that asset. An RFID reader with up to 5cm reading range is suitable for this job. However, since this part of the system is simulated using Packet Tracer; therefore, we used the only available reader for both functionalities.

The readers are simulated to be placed in various locations within the company's building. As we can see in figure, there is one reader inside the Storage Unit which will be responsible for the second scenario mentioned in the first paragraph. There are also readers at the door step of the Storage Unit, in the east and south hallways, and in the Building Entrance. These readers are what was supposed to be gate RFID readers which will do the job of the first scenario mentioned in the first paragraph. To help differentiate between the readers, each reader was given a different ID than the others. Furthermore, the first digits in the reader's ID represent the location of the reader, whereas the

last two digits represent the function of the reader based on the given scenarios in the first paragraph. We can see in figure readers in the Storage unit are given IDs “11100” and “11101”. The first three digits are the same since they are in the same location, but the last two digits are different. If last two digits were “00”, that means it is for the first scenario. Contrarily, if the last two digits were “01”, it is for the second scenario.

All the readers are connected to the SBC using an IoT custom cable and programmed to read a tag and that tag’s ID along with their given ID to the SBC. Figure shows a snippet of the code that will be run on all the readers, and as we can see, if the variable “found” is false, no tags have read, so we will initialize the variable “lastCardID” with 0 and send it as a string to the SBC through port 0 using the function “customWrite”. On the other hand, if the variable “found” is true, a tag has been read, so we will send the tag ID as a string concatenated with the reader code to the SBC using the same function mentioned earlier. The same code is run on all readers with the difference of the reader’s code.

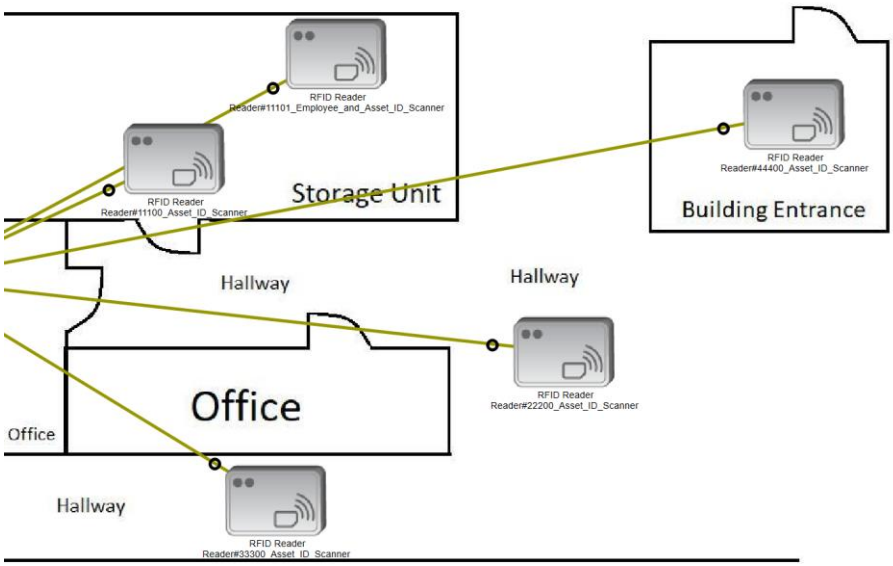


Figure 11: Readers in different locations

```
def loop ():
    global cardID, lastCardID, state
    devices = devicesAt(getCenterX(), getCenterY(), X_READ_DISTANCE, Y_READ_DISTANCE)
    found = False        # var found
    for i in xrange(0, len(devices)) :
        if devices[i] is getName():
            continue

        cardID = getDeviceProperty(devices[i], 'CardID')
        found = True
        break


    if not found:
        cardID = lastCardID = 0
        setState(2)
        customWrite(0, str(lastCardID))
        delay(100)
    else:
        if lastCardID != cardID:
            lastCardID = cardID
            sendReport()
            customWrite(0, "11101" + str(lastCardID))
            setState(0)
            delay(100)

    delay(DELAY_TIME)
```

Figure 12:Snippet of the code running on the readers

5.1.2 RFID Tags

Specifications and constraints

The RFID tags will be attached to assets and employee cards for the readers to detects them. Each tag has a unique ID to help identify different assets and employee cards from each other.

Specifications:

o Bandwidth: 850 – 950 MHz

o Latency:10 – 100 ms

o Accuracy: 10 – 50 $m2$

approximated tags location

o Security: plain

o Form factor: several designs with a size of $1 - 10\ cm3$ and weight around 3-10 g per tag

o Power consumption: passive RFID tags with 0 power consumption

Design and Verification

Since there is only one type of tag in Packet Tracer, we used it to represent both assets and employee cards. Each tag was given a different ID to differentiate each from the other. Figure shows the different tags we will be using in this system. The top row represents the employee ID cards, while the bottom row represents the assets IDs. We will store these IDs in a database in the cloud along with the asset or employee information.

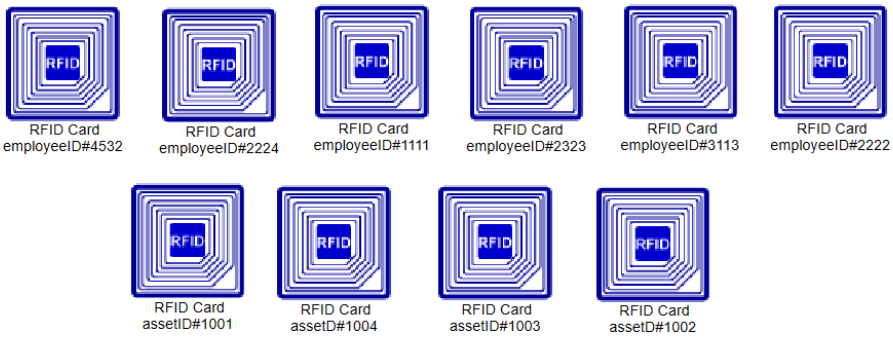


Figure 13: Different tags for both employees and assets.

5.1.3 Single Board Computer (SBC)

Specifications and constraints

The role of the SBC in this system is to send the received tag and reader IDs to the cloud server for processing and take actions based on the received response.

Design and Verification

The SBC is designed to be a connection point between the cloud server and Packet Tracer. Besides the RFID readers, we will connect other devices like an LCD and a siren to the SBC. Whenever a reader reads a tag, the SBC will receive the read tag's ID along with the reader ID. These IDs will be sent to a specific URL in the cloud server using an HTTP POST request method. The request message will also include a username and a password to authenticate the SBC to the server. Figure shows a snippet of the code which is in charge of communicating the tags read by the readers along with the reader's ID and the authentications information. We connected the all the readers to ports D0 to D4. Now, and using a for loop, we will continuously check on these ports. If the

read message which we stored it in the variable “x” is not a zero, then we create an JSON object which include the username, password, tag ID, and the reader ID. This object will then be sent using HTTP POST method to a specific URL in cloud server as mentioned above. The SBC will then for a response message from the server which will be handled in the function “onDone()”.

```
def main():
    username = "packet_tracer"
    password = "packet@123321"

    while True:
        for i in range(5):
            x = customRead(i)
            if x != "0":
                myobj = {"username" : username,"password" : password,"TagID" : x[5:], "ReaderID" : x[0:5]}
                http = RealHTTPClient()
                http.onDone(onHTTPDone)
                http.post(url, myobj)
                delay(3000)
```

Figure 14:Snippet of the code communicating with server on the SBC

After the server receives the request message from the SBC and processes the received data, it will send a response back to the SBC. Many different response messages could be obtained, each based on a specific action taken by the server. Figure shows the different responses that could be received and the actions will be taken based on these responses. As mentioned above, two LCDs and a siren are also connected to the SBC. The LCD that is placed in the admin's office along with the siren are connected to ports D6 and D8 respectively, while the LCD which resides along side the HF close range reader is connected to port D7.

```
def onHTTPDone(status, data):
    print("status: " + str(status))
    print("data: " + data)
    if data[0:5] == "Moved":
        customWrite(6, "ID #"+data[5:9] + " Moved \n"+data[9:] )
        customWrite(8, 1)
        print("Asset with ID " + data[5:] + " was moved without permission")
    elif data == "not employee id":
        customWrite(7, "Scan your\nEmployee ID!!")
    elif data == "not asset nor employee id":
        customWrite(7, "Not asset nor\nemployee ID")
    elif data == "scan asset":
        customWrite(7, "Great. Scan\nThe asset ID.")
    elif data == "no request for employee":
        customWrite(7, "No request for\nthe employeeID")
    elif data == "asset scanned":
        customWrite(7, "You are SET.\nTake the asset.")
    elif data == "scan employee id first":
        customWrite(7, "Scan employee\nID first.!!")
    elif data == "no request for asset":
        customWrite(7, "No request for\nThis asset ID!")
    elif data == "already scanned":
        customWrite(7, "Asset already\nscanned.")
    else:
        customWrite(6, 0)
        customWrite(7, 0)
        customWrite(8, 0)
```

Figure 15:Snippet of the code responsible for the responses received from the server on the SBC

Another SBC is needed to do https request every 24 hours. The purpose of this request is to check the deadlines in the borrowing table. There is a function for checking the deadline and need to be executed every time a request occurred. Another alternative to do this job is to do scheduled task in Django. We tried to do that, but an error occurs which could not be solved since it requires a model that is not in Windows. This issue has been mentioned in the section of problems and solutions. As shown in figure 11, a request will be done every 3600 seconds which is equivalent to one day.

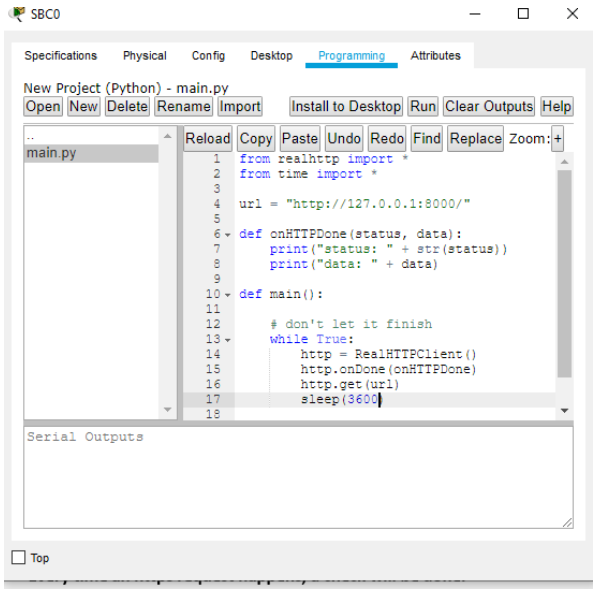


Figure 16: this is the scrips that will do http request every 24 hours.

5.2 Front-End

The **Django** Framework will be used to build the website application and it is a very popular framework. Django is based on Python and is a framework for both Frontend and Backend.

Design and Verification:

The front-end of this project will feature three main categories. The first category is the web interface for the admin of the service provider. The admin will do everything regarding giving authorization to the subscriber's admin through this interface. The second part is the subscriber's admin web interface, where the admin will add and delete both assets and users. The third category is the web interface for users. In this interface, users will be able to see all assets in the company and submit orders to borrow assets from the company.

The languages that will be used to design the front-end are HTML, CSS, and JavaScript.

5.3 Back-End

Heroku cloud application platform will be used to serve the back-end system. It provides to developers a practical way to deploy, operate and operate an entire application in the cloud. **Heroku** is a platform as a service (PaaS) which mean it provide significant number of services to the developers. Some of these components are servers, storage, virtualization, and middleware.

Design and Verification:

In **Heroku** website we are going to deploy **Django** application. **So, Heroku** back-end server will handle both requests and responses for the front-end and requests and responses sent from the SBC. Whenever the back-end server receives a request from the SBC with a tag ID, it will check whether the asset has been authorized to leave the premises or not. Based on whether the asset was requested to be borrowed or not, the asset's status will be updated by the server, and a notification will be sent. Therefore, the back-end server will keep company and its employee continuously knowing the assets' locations, and with who the assets are available.

In Heroku the selected programing language used for the back-end server will be Python, and the database is SQLite3.

5.4 Database

There are several types of databases that are available in the market each of them with different properties. In our project we choose SQLite because it is integrated within Django platform. So, it will be more rational to interact with it. There are some other choices like MySQL and PostgreSQL. In [5, Tab. 13], a comparison between these three databases. However, since SQLite is embedded in Django, it will be a good choice to go with.

Table 14: Comparison between databases [5]

Name	Advantages	Disadvantages	When to use	When not to use
SQLite	• File based • Standards-aware • Great for developing and even testing	• No user management • Lack of possibility to tinker with for additional performance	• Embedded applications • Disk access replacement • Testing	• Multi-user applications • Applications requiring high write volumes
MySQL	• Easy to work with • Feature rich • Secure • Scalable and powerful • Speedy	• Known limitations • Reliability issues • Stagnated development	• Distributed operations • High Security • Web-sites and Web-applications • Custom solutions	• SQL compliance • Concurrency • Lack of features
PostgreSQL	• An open source SQL standard compliant RDBMS • Strong Community • Strong third-party support • Extensible • Objective	• Performance • Popularity • Hosting	• Data Integrity • Complex, custom procedures • Integration • Complex designs	• Speed • Simple to sets up • Replication

Specification and constraints:

The database will be responsible for storing general information about the asset. That includes its whereabouts, status, ID, the person responsible for it in case of borrowing. The database will also be used to store employees and subscribers with their information.

Design and Verification:

- Requirement
  - There will be one administrator from system provider that going to trace several companies' subscribers.
  - Company can have several assets to be traced and several RFID readers to trace assets movement.
  - Company admin can specify allowed location for the assets.
  - Company subscriber and employees shall be able to locate asset easily with in company area.
  - Employees can get a permission to take the asset with them when they go out of the company.

Table 15: Conceptual and Logical Design

Conceptual Design	Logical Design

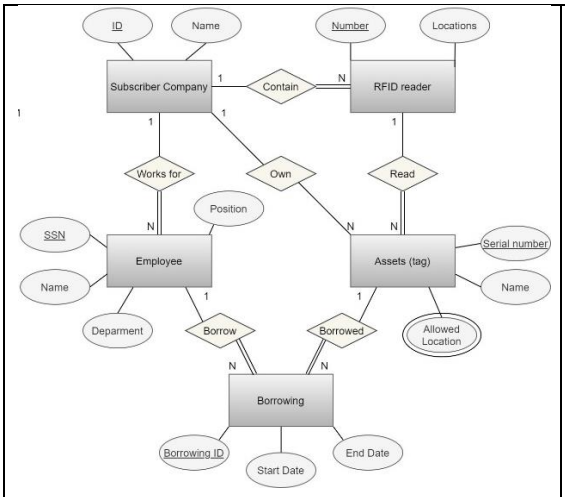


Figure 17:Conceptual Design

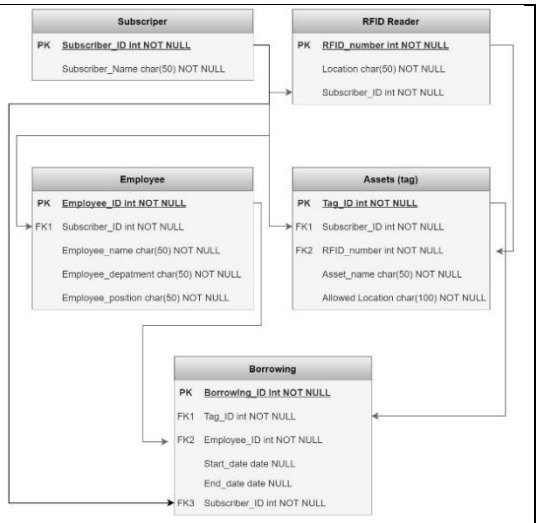


Figure 18: Logical Design

- **Implementation code:** https://github.com/OsamaBujwaied/Assets-Tracking/blob/main/assetstracking/models.py

6. INTEGRATION

6.1 Complete Prototype

- **Applications Web pages**
  - **Main page:** https://assets-tracking-coe.herokuapp.com/

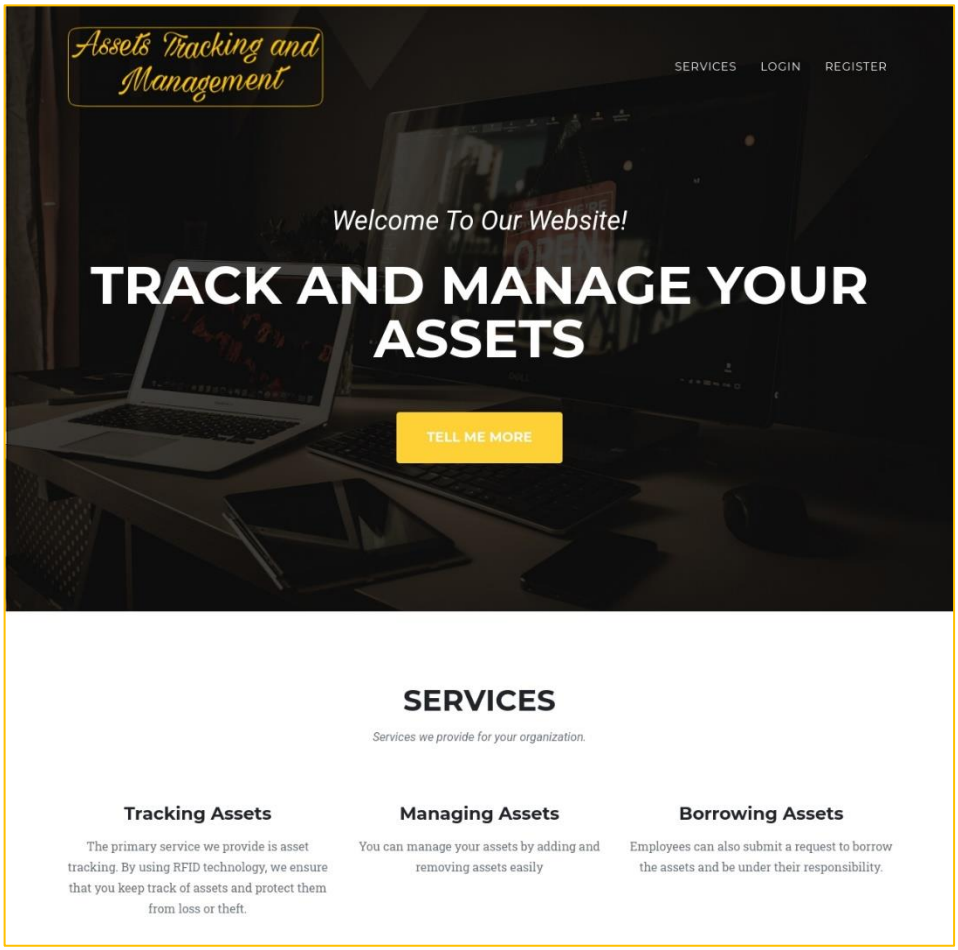


Figure 19:Welcome Page

this is the welcome page that will appear if the user enters the website. From this page, the user can login to the system as shown above in the right corner. Also, the new visitors can read about our services in this page.

System provider's page:

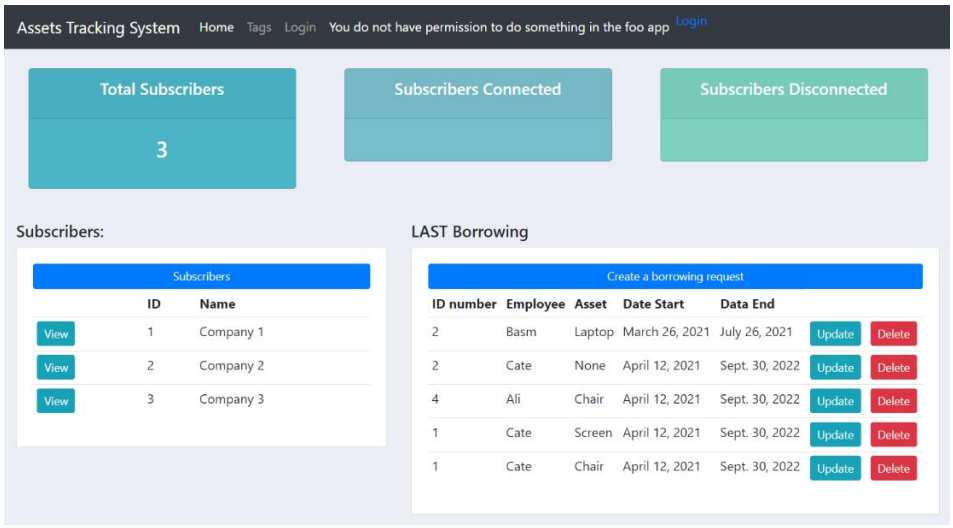


Figure 20:System Provider's Page

This page going to be used by us to trace and manage our company subscribers. It will be used to show the status of the last borrowing items and trace them.

- **Subscriber pages example:**

  https://assets-tracking-coe.herokuapp.com/subscriber/1/

  https://assets-tracking-coe.herokuapp.com/subscriber/2/

  https://assets-tracking-coe.herokuapp.com/subscriber/3/

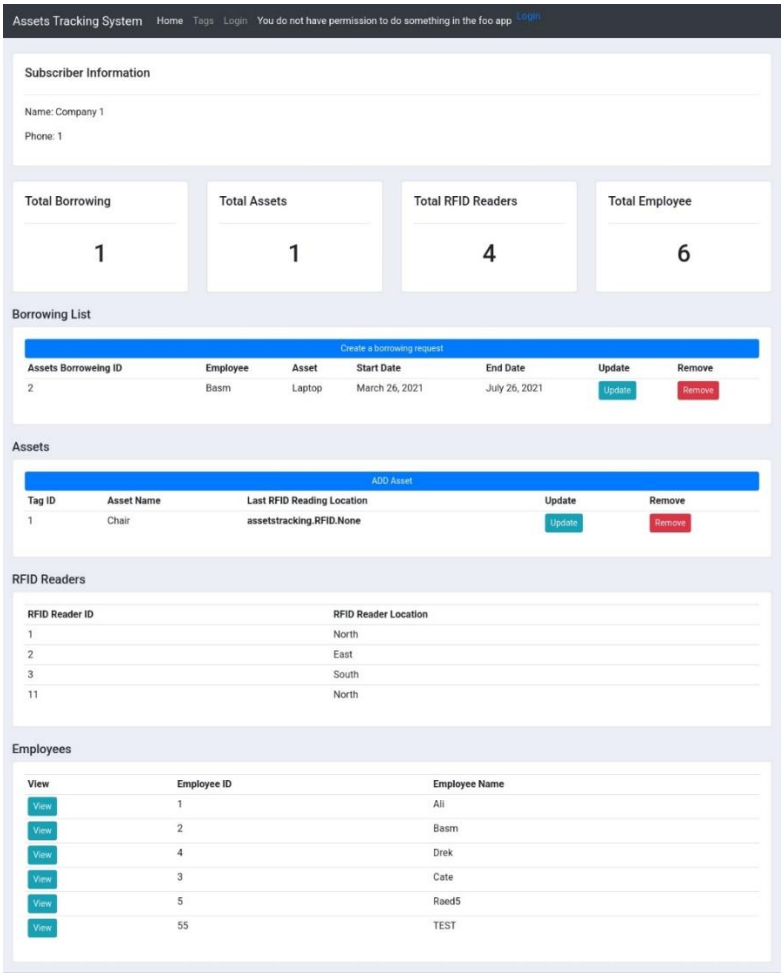


Figure 21:Subscriber's Admin Page

This page will be used by subscriber company administrator to manage and trace his company assets, RFID readers, employees, and borrowing. This page will provide 4 lists each list that contain most important information to the administrator.

- **Borrowing List:** will provide all the borrowed items in the company by which employee with start and end date of the borrowing. Administrator can create, extend, remove assets borrowing from the list.
- **Assets List:** it is a table that contains all assets that is registered to the system. It shows last location for the assets by using RFID

readers that distributed in the company. Also, subscriber administrator can add and remove and update assets by himself.

- **RFID Reader List:** This list going to show all RFID readers that going to be provided and installed by service provider. These readers will be distributed all around company building in a way that can capture assets movement wherever it goes. Subscriber company obviously will not be able to any modification to RFID readers.
- **Employee List:** a table that is contains every employee registered in a single subscriber (company) with his id. Also company administrator can view which assets were borrowed by each employee by pressing the **view** button.

- **Employees pages examples:**

  https://assets-tracking-coe.herokuapp.com/employee/1/

  https://assets-tracking-coe.herokuapp.com/employee/2/

  https://assets-tracking-coe.herokuapp.com/employee/3/

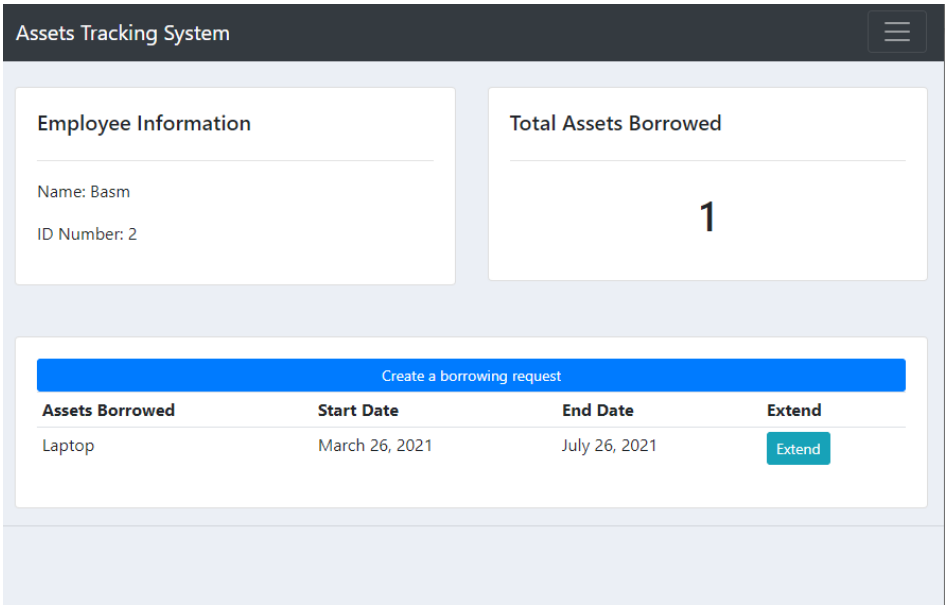


Figure 22:Employees Page

This page will appear to employees. In this page the employee will be able to see all the assets that he borrowed with start and end date of each of them as shown above. Employee can only request now borrowing and request extension of borrowing time of the assets.

- **Admin page:** https://assets-tracking-coe.herokuapp.com/admin/

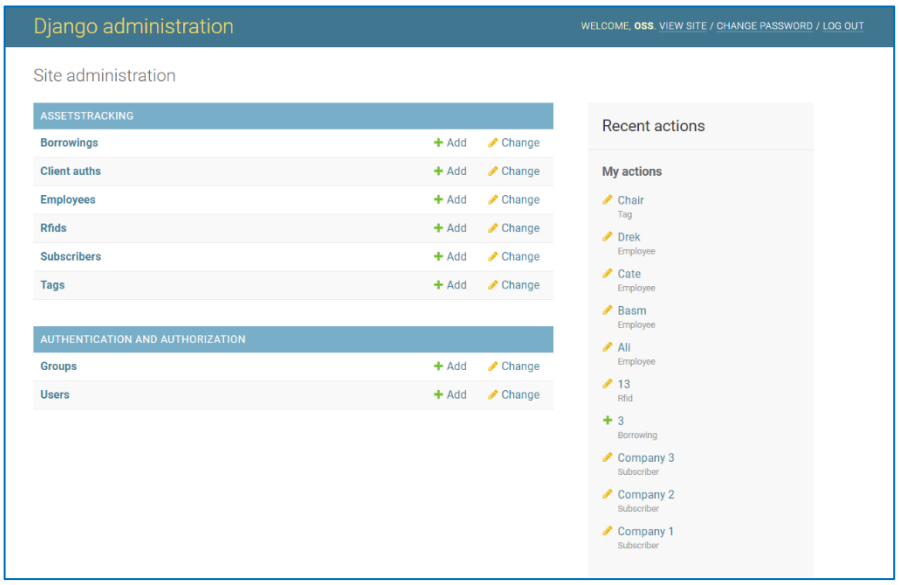


Figure 23:Admin Page

This page is dedicated for system provider only no one else can get to this page. It is in-built from Django web framework using SQLite database. System provider can use this page to access and manage all system information and users's account.

- **Tags page:** https://assets-tracking-coe.herokuapp.com/tags/

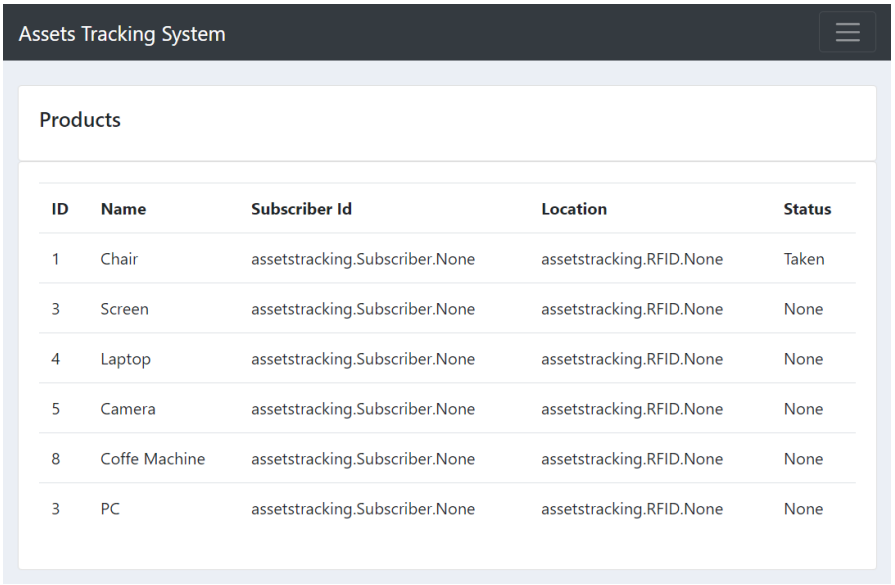


Figure 24:Tags Page

This page will used by everyone in the system to located assets, and to know their own status. The status can show you if the assets are available to be taken or not.

- **Login page:** https://assets-tracking-coe.herokuapp.com/assets/login/

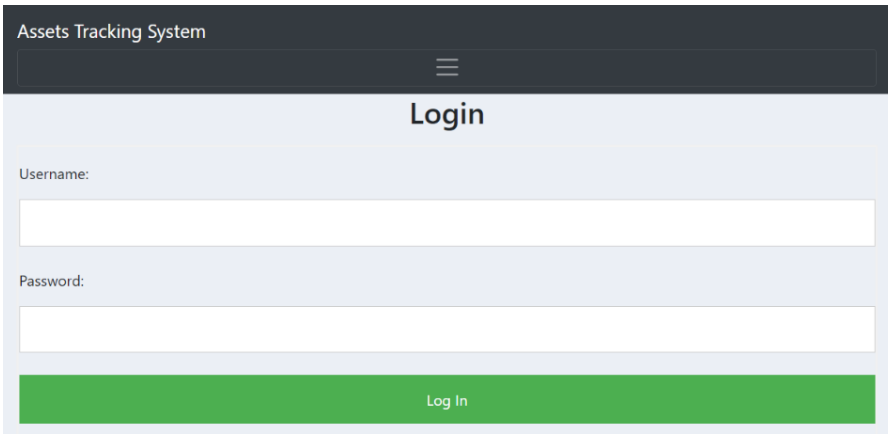
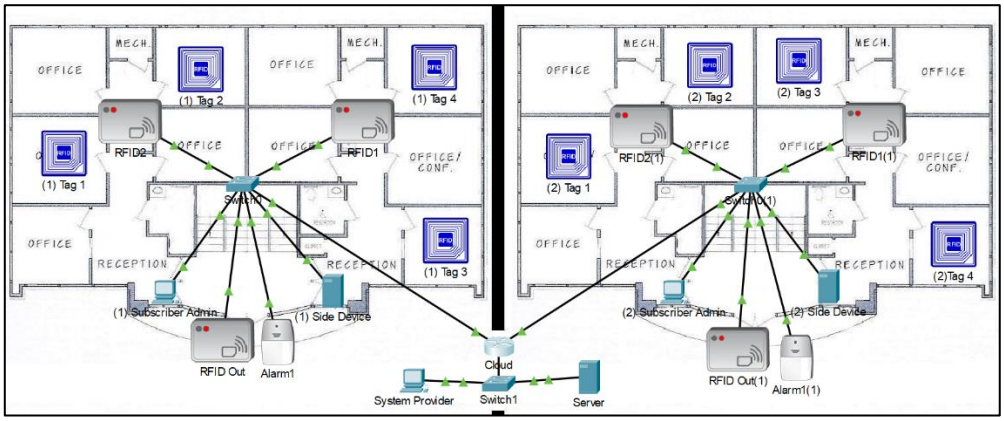


Figure 25:Login Page

This page will be available to subscriber administrator, and employee to be able to login to the system.

- **Packet Tracer:**

Figure 20 below will show illustration of the overall connection between the subscriber administrator, system provider, and Heroku PaaS.
The website application that was made by Django will be deployed using Heroku. Company's facility will be connected to the system using the Internet to manage and get all the needed information.



Each company will be provided with several long-range RFID. These RFID readers will be distributed by system provider in a way that they can track assets (tag) movement in every place in the company. Administrator of the companies will be provided with UHF passive RFID tags. Administrator can insert these tags to any assets he wants and then update its information in the database.

Figure 19 shows a packet tracer file page where all the simulated components connection that resides in the client side are connected together.

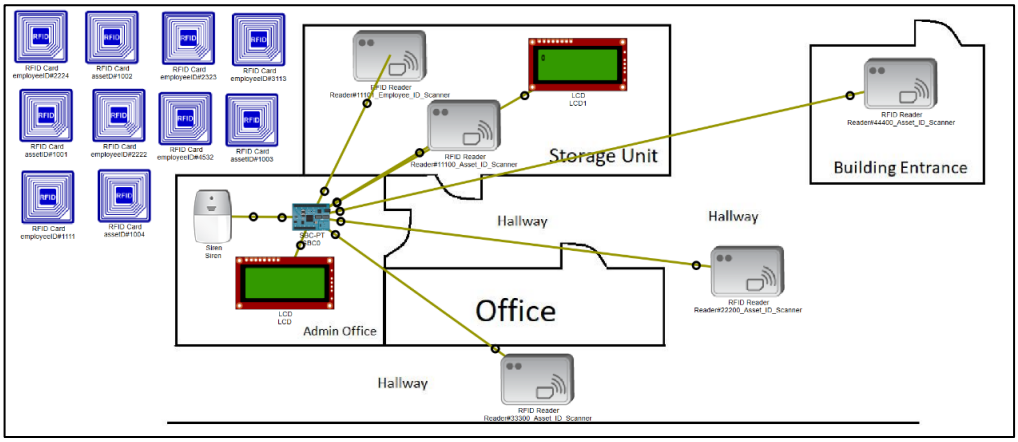


Figure 26: Packet Tracer File

6.2. Testing Results: this section will illustrate all use cases scenarios.

- **Login to the system:**

There are three types of authority:

1- **Service provider:** they own the service. Simply, they can do everything.

2- **Company Administrator:** a person that is responsible for assets tracking. This type can track, add, and delete assets. In addition, he has a list of employees' names, assets' tags, and readers.

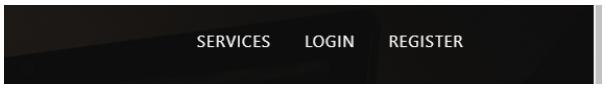
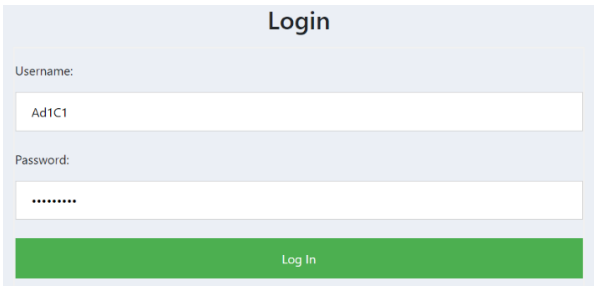
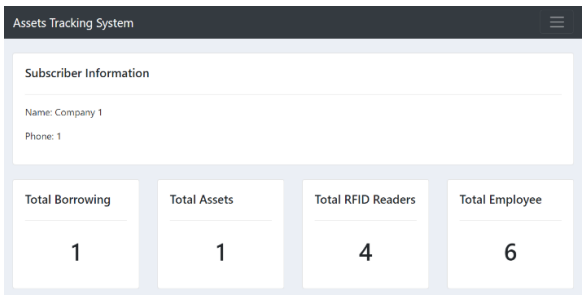
3- **Employees:** they only can request a borrowing, and see the tags

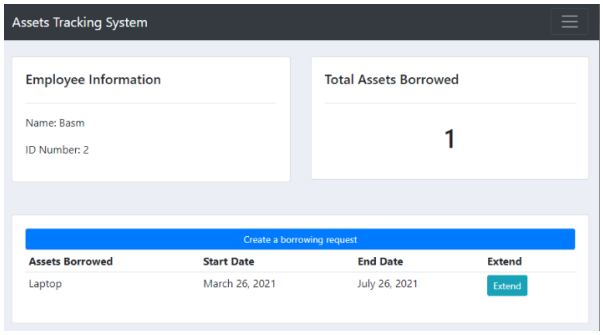
Let us try to login using username and password of the admin.

```
Username: em1C1
Password: Qw654321
Employee name and ID: Drek, 4

Username: em2C1
Password: Qw654321
Employee name and ID: Cate, 3

Username: em3C1
Password: Qw654321
Employee name and ID: Basm, 2

Username: Ad1C1
Password: Qw654321
Employee name and ID: Ali, 1
```

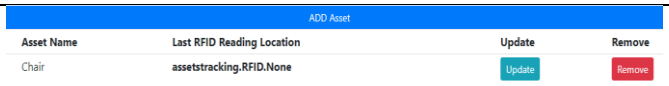
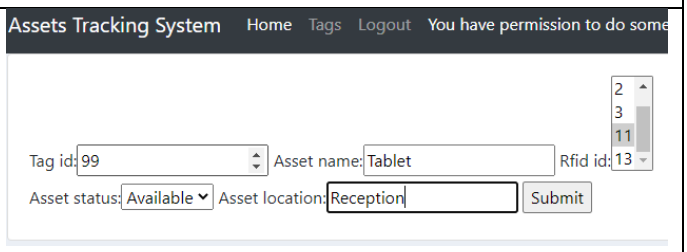
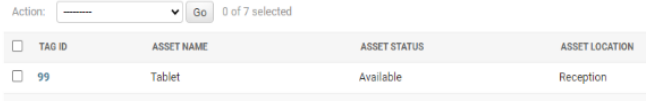
Figure 27: Usernames and passwords

Table 16: Login to admin and employee steps

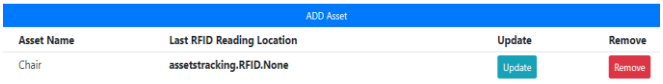
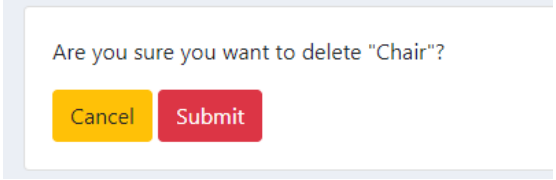
1- Click login:	 *Figure 28: Login 1*
2- Enter username & password:	 *Figure 29: Login 2*
3- The dashboard of the administrator will appear.	 *Figure 30: Login 3*

4- Do same steps but enter the username and password of an employee. Then the dashboard of the employee will appear.	

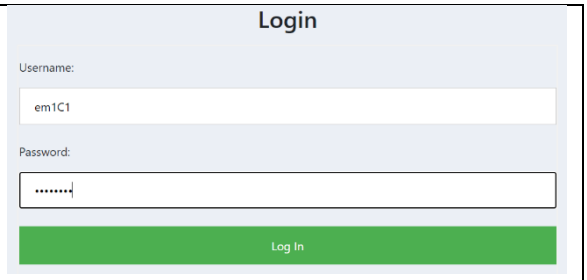
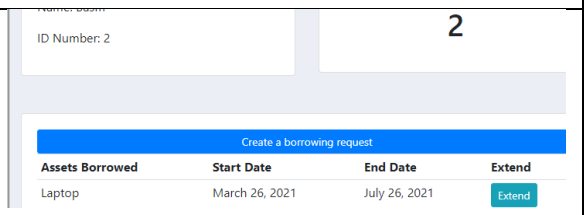
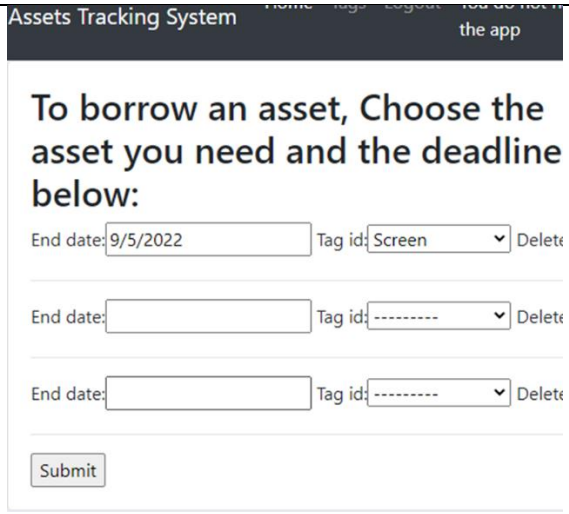
- **Adding Asset:**

1- Click on add asset.	 *Figure 31: Add asset 1*
2- Fill the information:	 *Figure 32: Add asset 2*
3- Now, we can see the asset has been added to the database.	 *Figure 33: Add asset 3*

- **Deleting Assets**

1- Click on Delete	 *Figure 34: Delete an asset 1*
2- Press in Submit:	 *Figure 35: Delete an asset 2*

- **Borrowing an asset**

1 - Login with an employee account	 *Figure 36: login page*
2 - Click on Create a borrowing request.	 *Figure 37:Create Borrowing request button*
3 - Choose the asset & determine the deadline.	 *Figure 38:Creating borrowing request page*

- **Scan the employee id.**

  After submitting a request for an asset, a record will be added to the database table of borrowings as show in figure 27. Then, the employee will go to the place where asset resides. The first thing to do before taking the asset is to scan the employee's card ID on the close-range reader. Figure 28 show the printed message on the LCD after scanning the employee ID. On the hand, figure 29 shows the fields "EMPLOYEE ID SCANNED" and "READER CODE" are set to 1 and 111 respectively. The "EMPLOYEE ID SCANNED" field is to indicate the scanned employee ID correspond to the requested employee ID, whereas the field "READER CODE" filed indicate the code that was given to the reader, which will help in making sure the employee ID and the asset ID are scanned from the same reader. This purpose of this step is to make sure that the employee who made the request is the same person as the one who is collecting the asset.

BORROWING ID	START DATE	END DATE	EMPLOYEE ID	TAG ID	EMPLOYEE ID SCANNED	ASSET ID SCANNED	READER CODE
1	April 20, 2021	May 9, 2022	Basm	Screen	0	0	0

1 borrowing

Figure 39:Borrowing request record added in the database.

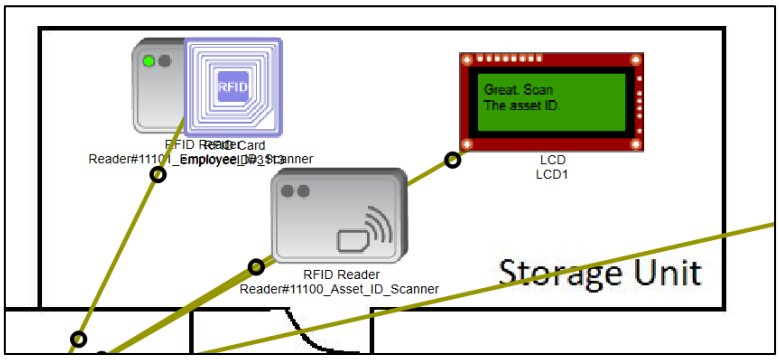


Figure 28:Message after successfully scanning employee ID.

BORROWING ID	START DATE	END DATE	EMPLOYEE ID	TAG ID	EMPLOYEE ID SCANNED	ASSET ID SCANNED	READER CODE
1	April 20, 2021	May 9, 2022	Basm	Screen	1	0	111

1 borrowing

Figure 29:Database record after successfully scanning employee ID.

- **Scan asset's tag.**

  The next step after scanning the employee ID, is scanning the asset ID. Figure 30 shows the message printed on the LCD after scanning the asset, whereas figure 31 shows the database record after scanning the asset. The field "ASSET ID SCANNED" is set to 1. The essence of this step is to first make sure the employee scanned the correct asset as in the borrowing request, and second help the gate readers placed at the doorstep and hallways recognize the scanned asset so they don't alarm the admin.

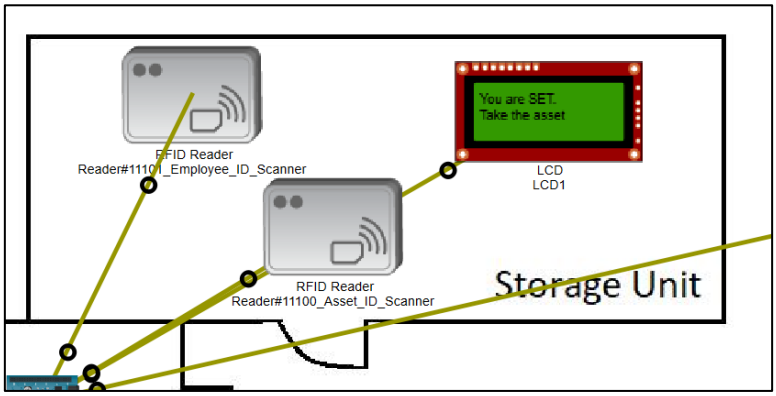


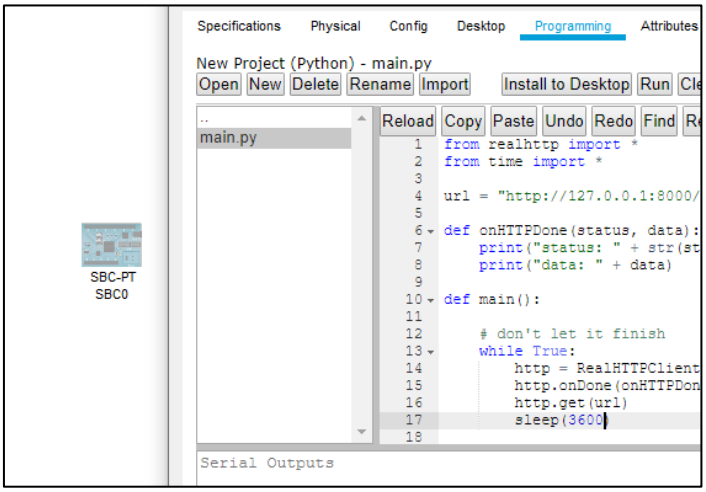
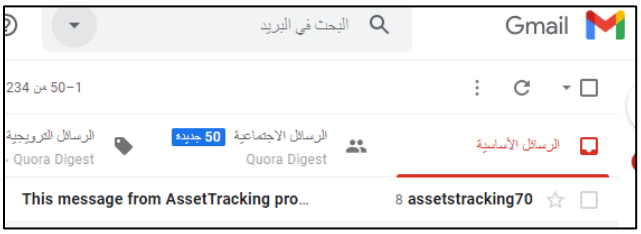
Figure 40:Message after successfully scanning asset ID.

BORROWING ID	START DATE	END DATE	EMPLOYEE ID	TAG ID	EMPLOYEE ID SCANNED	ASSET ID SCANNED	READER CODE
1	April 20, 2021	May 9, 2022	Basm	Screen	1	1	111

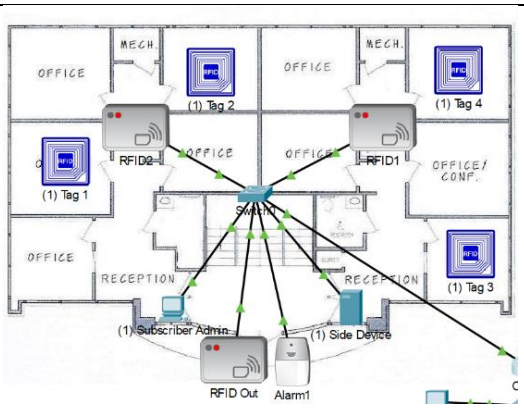
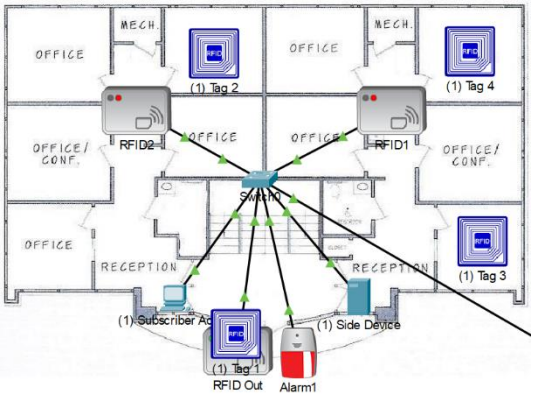
1 borrowing

Figure 41:Database record after successfully scanning asset ID.

- **Sending email to employee if he did not return borrowed asset before the deadline.**

1.To demonstrate this use case, we will create a borrowing request with a date which the deadline has passed.	 *Figure 42: Sending an email 1.*
2.Run the script that do HTTP request every 24 hours.	 *Figure 43: Sending an email 2.*
3. Check the email.	 *Figure 44: Sending an email 3.*

- **Warning if asset get out of the company.**

All the assets inside the company	
When some asset captured exiting the company. Alarm and siren will turn on and warning will be sent to administrator.	

6.3. Problems & Solutions:

1- Sending an email to employee if he did not return borrowed asset.

- **Problem:** Regarding sending an email if an employee did not return the asset. There is another way to do scheduled task using Django. But when we tried to do it, an error occurs. The error is as shown in Figure 35:

  After searching to solve this error, we found that module fct1 is not exist in Windows.

```
  File "<frozen importlib._bootstrap_external>", line 728, in exec_
  File "<frozen importlib._bootstrap>", line 219, in _call_with_fra
  File "C:\Users\ALOSTATH\Desktop\x\env\lib\site-packages\django_cr
    from django_crontab.crontab import Crontab
  File "C:\Users\ALOSTATH\Desktop\x\env\lib\site-packages\django_cr
    import fcntl
ModuleNotFoundError: No module named 'fcntl'

(env) C:\Users\ALOSTATH\Desktop\x\Assets-Tracking-main>
```

Figure 45: Scheduling error

- **Solution**:

  We accomplished this functionality by running SBC device that do HTTP request every 24 hours, in each request, a comparison will be executed to determine weather an email should be sent or not.

7. CONCLUSION

In summary, we have successfully built an asset tracking website with an easy-to-use user interface. Furthermore, we have used RFID technology for tracking assets. Since current semester is online, we could not integrate our work using hardware components. Instead, we have used a simulation program which is cisco packet tracer to demonstrate the functionality of our system. In addition, our service is deployed on Heroku cloud to make it accessible.

Lessons we learned: -

One of the most important lessons we learned is the importance of parallel work, so that the work of a member does not depend on other member's work. If this dependency occurs, there may be a delay in the delivery of requirements. Another lesson is the importance of periodic project evaluation and weekly meetings to discuss progress.

Suggestions for improvement: -

This project can be developed by building a mobile application to facilitate access to the service.

8. REFERENCES

[1] "Employee Theft Costs U.S. Businesses $50 Billion per Year", *Securitymagazine.com*, 2017. [Online]. Available: https://www.securitymagazine.com/articles/88432-employee-theft-costs-us-businesses-50-billion-per-year. [Accessed: 03- Mar- 2021].

[2] G. S. Khalsa, "An Inside Job," https://www.awci.org/. [Online]. Available: https://www.awci.org/cd/pdfs/9004_g.pdf.

[3] Heroku Security. Heroku. (2021, April). https://www.heroku.com/policy/security.

[4] Security in Django. Django. (2021, April). https://docs.djangoproject.com/en/3.2/topics/security/.

[5] Y. Wang, "SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems," *Medium*, 10-Aug-2017. [Online]. Available: https://medium.com/@yangforbig/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems-afd5afd6566. [Accessed: 21-Apr-2021].

9. APPENDIX

- **Assets Tracking website links:-**
  - **System Provider Page (Main):** https://assets-tracking-coe.herokuapp.com/
  - **Admin Database Page:** https://assets-tracking-coe.herokuapp.com/admin
  - **Welcome Page:** https://assets-tracking-coe.herokuapp.com/welcome
  - **Subscriber Example Pages**
    - https://assets-tracking-coe.herokuapp.com/subscriber/1/
    - https://assets-tracking-coe.herokuapp.com/subscriber/2/
    - https://assets-tracking-coe.herokuapp.com/subscriber/3/
  - **Employee Example Pages**
    - https://assets-tracking-coe.herokuapp.com/employee/1/
    - https://assets-tracking-coe.herokuapp.com/employee/2/
    - https://assets-tracking-coe.herokuapp.com/employee/3/
    - https://assets-tracking-coe.herokuapp.com/employee/#/
  - **Tag's page:** https://assets-tracking-coe.herokuapp.com/tags/
  - **Login page:** https://assets-tracking-coe.herokuapp.com/login

- **GitHub Program Link:-**

  https://github.com/OsamaBujwaied/Assets-Tracking
  - **Database Models code:** https://github.com/OsamaBujwaied/Assets-Tracking/blob/main/assetstracking/models.py
  - **URL Links code:** https://github.com/OsamaBujwaied/Assets-Tracking/blob/main/assetstracking/urls.py
  - **Views code** (connect URL to its HTML pages): https://github.com/OsamaBujwaied/Assets-Tracking/blob/main/assetstracking/views.py
  - **HTML pages code folder:** https://github.com/OsamaBujwaied/Assets-Tracking/tree/main/assetstracking/templates/assetstracking

- **Heroku** cloud application platform link**:** https://www.heroku.com

- **Django Tutorial by Dennis Ivy:** youtuber playlist used to learn Django https://youtube.com/playlist?list=PL-51WBLyFTg2vW-_6XBoUpE7vpmoR3ztO

- **Appendix for every member's work is in Teams**