Princess Sumaya جامعـــة
University الأميـرة سميّـة
for Technology للتكنولوجيا

Princess Sumaya University for Technology King Hussein

School of Computing Sciences

BON-Parking System

By:

Omar Khaldoon Khwaileh 20180170

Bakir Mahmoud Al-Khatib 20180583

Nassri Raed Attoun 20180721

Supervised by:

Dr. Hazem Qattous & Dr.Abedalrhman

This is the project submission for Graduation Project two Course for the degree of bachelor in Software Engineering.

Summer Semester, 2021-2022

Acknowledgment

Words only cannot express our thanks and gratitude to our parents, friends, and to everyone who supported us all the way to become the people that we are today.

We also would like to thank our supervisors Dr. Hazem Qattous & Dr. Abedalrhman for their full support, and encouragement through our Project duration.

We appreciate all of our Drs efforts, work, and time given to make us excel in what we do, especially our wonderful, department.

Last but not least, we want to thank ourselves

Table of Contents

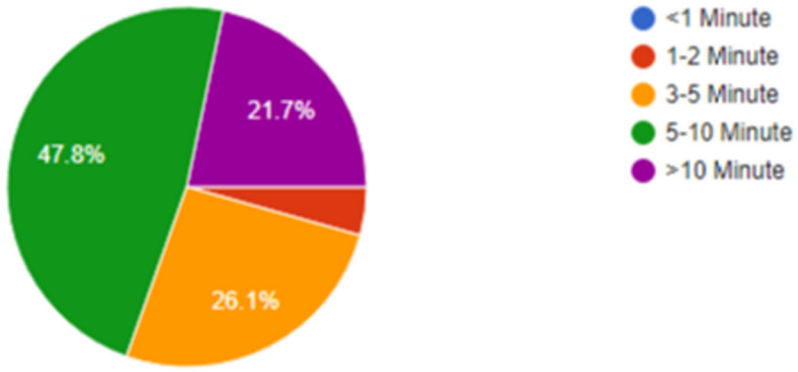
Chapter One: Introduction

This section describes the problem statement and what leads us to create our project, the problem motivation, and a comparison between other projects and what makes our project unique from other projects.

1.1 Problem Statement

Tired of always getting late by 2-5 Minutes? Sick of spending much time searching for a proper place to park? Always searches in circles around the park? What a headache, keep smiling that's what BON E-Parking is aiming to help you in.

According to a survey has been done, it says the majority of the student and faculty members face a problem with finding a spot for their cars, it takes them 5-10 Minutes an average to find a spot, while others take more than 10 minutes, what's a big-time being wasted!



Even those who don't face a problem with finding parking, think it would be nice to have an application that would make things easier.

1.2 Related work

Based on our research, there were multiple projects related to the problem that BON E-Parking is aiming to solve. The current system uses Arduino [7] along with a card reader to let only authorized people enter the parking [1]. This solution may be out of the scope of our project, as our project is limited to a mini parking area.

Another solution similar to our project scope uses Arduino and sensors [3] but it doesn't contain any related network connection field. Other better solution used Ethernet-based connection [2] to connect needed devices to the parking network, but it has drawbacks since it requires cables. Having a wirelessly based connection using wife modules [6] along with the Arduino [8]will make it easier in installation and more feasible as what BON E-Parking aims to do.

Most commonly infrared sensors have been used for this kind of project [3], but it gets affected by light sources. As other better solution uses ultrasonic sensors [5], placing them on the top of the parking [4], any change in the distance between the surface and the celling means the garage is busy. But since our project will be applied to open parking, it won't be possible to place them upside the car spot, which will require us to place them downside the cars.

Using ultrasonic sensors, to make sure that the availability of each spot is calculated based on the distance, as well as placing the sensors in the middle of each car spot to make sure that the availability result is valid, without leaving any possibility of system wrong indicators.

The system is a combination of the hardware and software, to create a complete module by Placing sensors and Arduinos, in the middle of each car parking connecting them all wirelessly, to the institution network. Using a proper GUI that shows the availability of each parking, indicating them with colors. This will help the driver to see the nearest available park, which in its turn will reduce parking search time.

1.3 Project Purpose and Objectives

The main project aim is to become one of the best solutions for crowded parking places, That considers time as a very precious factor in human lives, and that every moment is valuable. Make it easier to find a park without going in circles and circles.

The project intends to deliver an easy, simple, fully working android app that does all of the required functionality goings through the entire development process and covers all design phases and development.

1.4 Problem Motivation

Our Passion comes from our need since many students and faculty members are facing this issue without any available solution. Both saving time, and making parking easier is our fire flame and what motivate and give us the power to accomplish our project.

The impacts of creating this project will be huge especially for crowded Parking, whether it’s time or organizational-related.

Chapter Two:

Software Project Management Plan (SPMP)

2.1 Introduction

2.1.1 Project Charter

Project Title: BON E-Park
Date of Authorization: Oct, 10, 2021 **Project Start Date:** Oct, 10, 2021 **Projected Finish Date:** June, 20, 2021
Key Schedule Milestones: • Complete the first version of the Software Requirement Specification (SRS) by Jan, 22, 2021 • Complete a Function version of the software by May, 1, 2021 • Complete Testing Phase by May, 15, 2021 • Deliver Complete documentation and System by June, 20, 2021
Budget Information: 170JDS have been allocated for the project, the majority of the budget will be spent on sensors and Arduinos as well as hosting the application and programs license. Additional funds may be provided if needed.
Project Manager: Omar Khwaileh, (00962) 787878-492, OMA20180170@std.psut.edu.jo
Project Objectives: The main objective of this project is to build a fully working version of the software and complete documentation by the specified deadlines.
Main project success criteria: Software reaches all expectations by going through the entire development process and getting a well-tested program that meets all specifications delivered on time.
Approach: • Develop a clear work breakdown structure (WBS) within the first three weeks to make sure everything goes as planned • Within the First four months complete a clear software requirement specification (SRS) to ensure clear project scope • As the project goes on hold a weekly meeting with the project team and supervisor to review the progress

Roles And Responsibilities

Name	Role	Position	Contact information
Dr.Abedalrhman	Supervisor	Supervise	A.lkhateeb@psut.edu.jo
Dr. Hazem Qattous	Supervisor	Supervise	H.qattous@psut.edu.jo
Omar Khwaileh	Project Manager, Team member	Manager and Developer	Oma20180170@std.psut.edu.jo
Baker Al-Khatib	Team member	Developer	Bak20180583@std.psut.edu.jo
Nassri Attoun	Team member	Developer	Nas20180721@std.psut.edu.jo

2.1.2 Project Scope Statement

Project Title: BON E-Park

Project Justification: The disturbance that most people face while searching for a proper place to park in, what a general issue that drivers and parking owners encounter!
Finally the need for the product to solve all of that justifies our project.

Product Characteristics and Requirements:

1- The system shall enable users to see the nearest available park.
2- The system shall enable users to see the number of available parks.
3- The system shall tell the user if the place is already crowded or not.
4- The application owner shall be able to add services.
5- The software shall be easy to use.

• For more detailed information about system requirements, see software requirements specification (SRS) chapter in the document.

Project Deliverables: Project charter, Scope statement, Cost management, work breakdown structure (WBS), Risk management, Software requirement specification (SRS), software design specification (SDS), Software Implementation, and software Testing Document (STD).
All will be covered in other sections in the document**.**

Product Deliverables:

- Android application: fully working application
- Database: a well-designed database includes all related info
- Back-end server: connects the application with Internet service and Database
- Rollout: Software will be published in the Play Store at the end of the project

Acceptance Criteria: Our goal is to complete and deliver the project within the specified project duration with the allocated budget, which gets customer satisfaction.

2.2 Assumptions and Constraints

Assumptions:

- The team has the required knowledge and experience to complete the project, including (Programming languages: Android, Java - Documentation, and back-end development)
- University will provide needed help regarding device installation at its parking.
- The established budget is sufficient to complete the project.
- The prototype building is funded by the university.
- The time is enough to complete the project.

Constraints:

- **Time:** We are constrained by the agreed-upon time, and the project must deliver before the deadline.
- **Customer Satisfaction:** The final project should satisfy the needs of our users.
- **Scope:** We adhere to the specified and approved project scope.
- **Cost:** The project budget is limited; we will try to stick with it.

2.3 Risk Management and Assessment

2.3.1 Stakeholder Analysis

Position	Role	Internal/ External
Project Supervisor	Supervise the project through the entire project duration, by holding scheduled weekly-meeting. To make sure everything is going as planned	Internal
Development manager	Implement and direct development team members, and ensure project needs are met	Internal
Project Manager	Monitor and manage the project, by making sure everything going within the defined schedule, scope, and cost	Internal
Development team	Implement the program, follows development manager instructions	Internal
Customers	Provide the needed budget	External
Users	Use the system	External

2.3.2 Risk Elicitation

Risks were found through brainstorming sessions, between team members along with the project supervisor throughout the project, as well as documentation analysis issued, to analyze the problems that faced other similar projects, in order to avoid and overcome them.

2.3.3 Risk Identification

In this section, the project team members list different categories of the risks that could face/occur, during the project execution period.

Risks		
Risk ID	Risk Description	Categories
R 01	The project team member can't make a face-to-face meeting due to COVID-19	Organizational
R 02	Team member requires more time to complete learning the required skills	Organizational
R 03	The project team isn't experienced with a proper documentation tool	Organizational
R 04	Using the community tools, the risks of not including needed features	Technical
R 05	The risk that users will find the application difficult to use	Product
R 06	Lack of qualified workers to do the work well or on time	Organizational
R 07	The possibility of not achieving customer satisfaction	Product
R 08	The risk of failing to deliver on the agreed service	Product
R 09	The risk of not completing the project on time	Organizational
R 10	The risk of producing a similar project	Business
R 11	The allocated budget is not enough	Organizational
R 12	Work starts before approval	Organizational

2.3.4 Risk Analysis

After risks have been listed, the project team will analyze each risk from point of impact and the probability of occurrence, which will define the priority of each risk, which will help in later stages in a way of handling the risk and how to respond to it.

Probability	
High	Has a very high probability to occur more than 60% (but not 100%).
Medium	Has a chance between 31% and 60% of occurrence.
Low	Has a Low probability to occur less than 30%.

Impact	
High	The project can't proceed with the project activity unless it is solved immediately.
Medium	Cannot proceed further with the project unless it's solved.
Low	Need to be solved or can be resolved at a later stage in the project or take another alternative.

Risk Analysis			
Risk ID	Probability	Impact	Priority
R 01	High	Medium	High
R 02	High	High	High
R 03	High	High	High
R 04	Medium	low	Medium
R 05	Low	Medium	Medium
R 06	High	High	High
R 07	Medium	High	High
R 08	Medium	High	High
R 09	Medium	Medium	Medium
R 10	Low	Low	Low
R 11	High	Low	Medium
R 12	Low	High	Medium

2.3.5 Risk Metric

Probability / Impact	Low	Medium	High
Low	**R 10**	**R 04**	**R11**
Medium	**R 05**	**R 09**	**R 01**
High	**R 12**	**R 07** **R 08**	**R 02** **R 03** **R 06**

2.3.6 Risk Response

Lastly, this part shows the counter-measure/response plan for each risk, to get a clear vision of the action needed to be taken and to be prepared for any consequences.

Risk ID	Description	Response
R 01	Find a proper tool/Platform to ensure communication with team members.	Reduction
R 02	Provide Extra time to learn the required skills, as well as use the Winter Break time as a learning period.	Mitigate
R 03	The Project Manager searches across platforms and finds that the “Latex “language has been used for well decimation, using the “Overleaf” tool.	Mitigate
R 04	Searching for a new tool that provides the needed features, as well as buying a program license is an available option.	Avoidance
R 05	Involve building application stage with Human-Computer Interactions Principles, which increase the level of usability of our program	Mitigate
R 06	Made policies for appropriate education and well training, to gain the required skills in a way that ensures high quality.	Reduction
R 07	Ensure the customer evolvement in all project stages, and by making prototypes.	Mitigate
R 08	Set clear deliverables and focus on core features.	Reduction
R 09	The project manager will change the working schedule in a way that increases team members' performance	Mitigate
R 10	Market monitoring, as well as System Upgrades and Updates.	Accept
R 11	The project manager will keep his eyes on the costs, and make a Budget management plan, as our budget is flexible needed money can be provided.	Reduction
R 12	Waits for the formal approval.	Mitigate

Chapter Three:

Software Requirements Specifications (SRS)

Requirements Versions

Date	Version
10/Dec/2021	1.0
10/May/2022	2.0

3.1 Requirements Elicitation Technique

3.1.1 Brainstorming Sessions

During the project period, team members set down together frequently as much as required, discussing ideas and new features related to the project, also it helped us find solutions to the problems that occurred, and so on.

Moreover, multiple brainstorming sessions were conducted with the project supervisor to discuss precipitated project turnover.

Brainstorming sessions were the main requirement elicitation technique that we used to gather the requirements.

3.1.2 Document Analysis

Document Analysis is a technique used to gather requirements during the requirements elicitation phase of a project. It describes the act of reviewing the existing documentation of comparable business processes or systems to extract pieces of information that are relevant to the current project and therefore should be considered project requirements.

Document analysis was conducted at the beginning of the project to study existing documents and projects close to our project idea, which led us to expand our perceptions and understand the differentiation aspects of our project.

3.1.3 Survey

At the begging of our project, a survey has been made and sent to the student and faculty members, to prove the problem and gather requirements from a large group of people, which gave us a clear vision of the problem and lead us to find the best suitable solution.

3.2 Product Requirements

Our system is a combination of hardware along with software, moreover, our system will have mainly two requirements types:

1. Product Hardware Requirement
2. Product Software Requirement
    2.1. Client Account.
    2.2. Admin Account.

3.2.1 Product Hardware Requirements

HR-1 **The System should be practical.**

1.1 The system should use as few wires as possible.

1.2 Hardware components should group up in one shape.

1.3 Hardware shape should be covered with 3D-Printing -Thermosetting Plastics-.

HR-2 **The system mainboard should be able to interact with the local Wi-Fi network.**

2.1 The system should be able to connect to a Wi-Fi network.

2.2 The system should be able to send Wi-Fi signal/packet.

HR-3 **The system mainboard should be ESP32.**

3.1 The system mainboard has a built-in Wi-Fi chipset.

3.2 The system mainboard should have the required pins.

3.3 The system mainboard shall fit in any spot.

HR-4 The system should have a ‘400 Tie Points Bread Board’

4.1 The system should have a main connection point.

HR-5 The system should include a distance measuring chipset.

5.1 The System should be able to measure the distance

HR-6 The system distance measuring Chip is ‘HC-SR04’

6.1 The System uses an Ultrasonic sensor to measure the distance

6.2 Ultra Sonic trigger pin send signal then wait for it to back to echo pin

HR-7 The system's main power source should be portable.

7.1 The system's main power source shall be a rechargeable battery.

7.2 The battery's main charging source is the sun.

Interdependencies

Requirements No	Refines To	Requires	Increase Cost Of	Increases Value Of
HR-1	Non	Non	Non	Non
HR-2	Non	Non	***HR-1***	***HR-1***
HR-3	***HR- 2***	Non	Non	Non
HR-4	Non	Non	Non	Non
HR-5	Non	***HR-4***	Non	Non
HR-6	***HR-5***	Non	Non	Non
HR-7	Non	Non	Non	***HR-1***

3.2.2 Product Software Requirements

Functional Requirements

3.2.2.1 Users Account Requirements

UFR-1 **The system shall enable users to access their accounts by asking them to provide:**

1.1 **E-mail**: the email of the application must be the same as the university email account.

1.2 **Password**: The password of the application must be the same as the email account password.

UFR-2 **The system should allow only authorized users to access the application.**

2.1 The user should be logged in successfully to get access.

2.2 The system main page should be the login page.

UFR-3 **The system shall validate user log-in coordinates.**

3.1 The system should include a hash function.

3.2 The system should generate a hash value for the provided password and compare it with the sorted hash value in the database.

3.3 The system should use the 'Firebase' hashing algorithm.

UFR-4 **The system shall enable the users, to see the whole parking area.**

4.1 The system interface should enable the users to see the whole parking area.

4.2 The system parking area should be similar to the reality.

UFR-5 **The system should use colors to indicate parking slot status.**

5.1 Humans are familiar with ‘traffic light’ lights.

5.2 The system should refine to ‘Available’ slots with a green light indicator.

5.3 The system should refine to ‘Occupied’ slots with red light indicator.

5.4 The system should refine to ‘device problem’ with a grey light indicator.

UFR-6 **The system shall show to the users the number of available parking slots.**

6.1 The system should count the number of occupied parking by counting the number of ‘available’ indicators parking slots.

UFR-7 **The system should notify the users of the limited number of available parking slots.**

7.1 The system should change the colors of the ‘available slots’ textView to ‘Red’ to indicate the limitations of available parking slots.

UFR-8 The system shall have a customizable theme.

8.1 The system should include a dark theme as well as a Light theme.

8.2 The system 'Light Theme' should be with white background and dark text.

8.3 The system 'Dark Theme' should be with a black background with light text.

UFR-9 The system should include a security man account.

9.1 The security man account was built for the security guards.

9.2 The security account will show the whole parking area to enable them to see available slot places.

9.3 The security account will help security guards to guide university guests to available slots.

UFR-10 The User should be able to change the account password.

10.1 The system should enable users to change their account passwords.

10.2 The user will be asked to provide the new password in a text field.

10.3 The user will be asked to repeat the provided password.

10.4 The user's new password cannot be less than six-character

UFR-11 The User should be able to reset the account password.

11.1 The system should enable users who forget their passwords to reset them.

11.2 The system shall reset the user password by sending an email confirmation to the provided account address.

11.3 E-mail should include a link to reset the user password

3.2.2.2 Admins Account Requirements

AFR-1 **The system shall enable users to access their accounts by asking them to provide:**

1.1 **E-mail**: The e-mail of the application must be provided by the university's IT department.

1.2 **Password**: The password of the application must be provided by the university's IT department.

AFR-2 **The system should allow only authorized users to access the application.**

2.1 The Admin should be logged in successfully to get access.

2.2 The system main page should be the login page.

AFR-3 **The system shall validate user log-in coordinates.**

3.1 The system should include a hash function.

3.2 The system should generate a hash value for the provided password and compare it with the sorted hash value in the database.

3.3 The system should use the 'Firebase' hashing algorithm.

AFR-4 The system shall enable the Admins, to see the whole parking area

4.1 The system interface should enable the users to see the whole parking area.

4.2 The system parking area should be identical to the reality.

AFR-5 The system shall show to the Admins the number of available parking slots.

5.1 The system should count the number of available parking by counting the number of ‘available’ indicators parking slots.

AFR-6 The system shall show to the admin the number of occupied parking slots.

6.1 The system should count the number of occupied parking by subtracting the number of ‘available’ indicators from the total number of parking slots.

AFR-7 The system should include a register screen.

7.1 The Admin account should be able to register a student who has a car entry permit.

AFR-8 The admin register page shall validate provided info

8.1 First name: the first name text field is a required non-empty field min user first name character is 3.

8.2 Second-name: the second name text field is a required non-empty field.

8.3 Email: email should be formed as a PSUT student account form

8.4 Plate: plate text field is a required non-empty field.

8.5 Phone number: phone number is required with validation as Jordanian Num

Non Functional Requirements

NFR-1 The System should be practical.

1.1 The system application should be available for all students.

1.2 The system application should be available for IOS and Android users.

NFR-2 The System application development language should be Flutter.

2.1 According to 'BON' Company regulation, the application development language is Flutter.

NFR-3 The System should interact with Firebase Database.

3.1 According to 'BON' Company regulation, the system database is Firebase.

NFR-4 The system GUI should be easy to use.

4.1 The System usability should Follow Ben Shneiderman's Rule of HCI.

4.2 The system should enable the users to get to use the application for a maximum of 3 minutes.

NFR-5 The system database should be secured.

5.1 The system database coordinates should be secured by storing secret coordinates as hash values – not clear text –.

Interdependencies

Requirements No	Refines To	Similar To	Requires	Increase Cost Of	Increases Value Of
Users Account					
UFR-01	Non	Non	Non	Non	Non
UFR-02	***UFR-1***	Non	Non	Non	Non
UFR-03	***UFR-2***	Non	Non	Non	Non
UFR-04	Non	Non	Non	Non	Non
UFR-05	Non	Non	Non	Non	Non
UFR-06	Non	Non	Non	Non	Non
UFR-07	Non	Non	Non	Non	Non
UFR-08	Non	Non	Non	Non	Non
UFR-09	Non	Non	Non	Non	Non
UFR-10	Non	Non	Non	Non	Non
UFR-11	Non	Non	Non	Non	Non
Admin Account					
AFR-01	Non	***UFR-1***	Non	Non	Non
AFR-02	Non	***UFR-2***	Non	Non	Non
AFR-03	Non	***UFR-3***	Non	Non	Non
AFR-04	Non	***UFR-4***	Non	Non	Non
AFR-05	Non	***UFR-6***	Non	Non	Non
AFR-06	Non	Non	***AFR-5***	Non	Non
AFR-07	Non	Non	Non	Non	Non
Non-Functional					
NFR-01	Non	Non	Non	Non	Non
NFR-02	Non	Non	Non	Non	***NFR-1***
NFR-03	Non	Non	Non	Non	Non
NFR-04	Non	Non	Non	Non	Non
NFR-05	***NFR-3***	Non	Non	Non	Non

3.3 Requirement Ranking

3.3.1 Ranking Aspects

penalty		Points
High	The project will be unusable if we proceed without implementing it.	3
Medium	It may affect the project in the future in case we didn't implement it.	2
Low	The project may be done without it.	1

Importance		Points
High	It's an essential requirement the project can't be done without it.	3
Medium	It has high importance to be included in our project.	2
Low	We can go without it.	1

Time		Points
High	This requirement will take too much time to implement.	1
Medium	It needs time to be done.	2
Low	It doesn't require any time.	3

3.3.2 Ranking Analysis

Ranking Analysis				Cumulative Points
Req ID	**Penalty**	**Importance**	**Time**	----------
UFR-01	High	High	Medium	8
UFR-02	High	High	Medium	8
UFR-03	High	High	Medium	8
UFR-04	High	High	High	7
UFR-05	High	High	Medium	8
UFR-06	Low	Medium	Low	6
UFR-07	Low	Medium	Low	6
UFR-08	Low	Medium	Low	6
UFR-09	High	High	High	7
UFR-10	High	High	Medium	8
UFR-11	Low	High	Medium	6
AFR-01	High	High	Medium	8
AFR-02	High	High	Medium	8
AFR-03	High	High	Medium	8
AFR-04	High	High	High	7
AFR-05	Low	High	Low	7
AFR-06	Medium	High	Low	8
AFR-07	High	High	Medium	8

3.3.3 Requirement Ranking

According to the above analysis, requirement rank has been measured using three different aspects:

- **Time**: the time each requirement is expected to take.
- **Penalty**: the level of penalties that may occur in case of not implementing a specific requirement.
- **Importance**: The importance of each requirement to the system.

Rank one (8 Points):

- UFR-01
- UFR-02
- UFR-03
- UFR-05
- UFR-10
- AFR-01
- AFR-02
- AFR-03
- AFR-06
- AFR-07

Rank Two (7 Points):

- UFR-04
- UFR-09
- AFR-04
- AFR-05

Rank Three (6 Points):

UFR-06

UFR-07

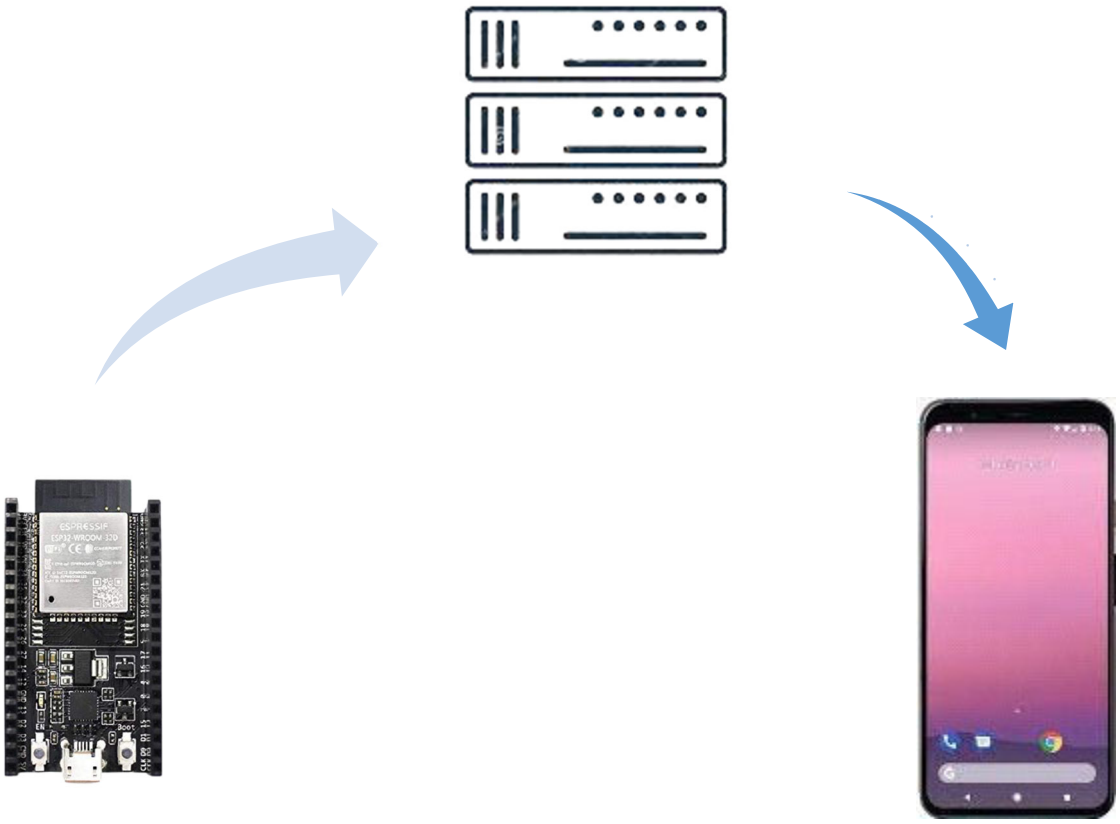
UFR-08

UFR-11

Chapter Four:

Vertical Prototype

4.1 Mapping



4.2 Explanation

4.2.1 ESP 32

The ESP32 measures the distance between it and the nearest object, based on a simple distance equation it decides the availability status of the parking spot and using the pre-defined port and IP address it sends the availability result to the local python socket server with defined coordinates.

4.2.2 Python Socket Server

In its turn, the python socket server receives and handles the signal, and resends it to the mobile phone application using TCP Protocol.

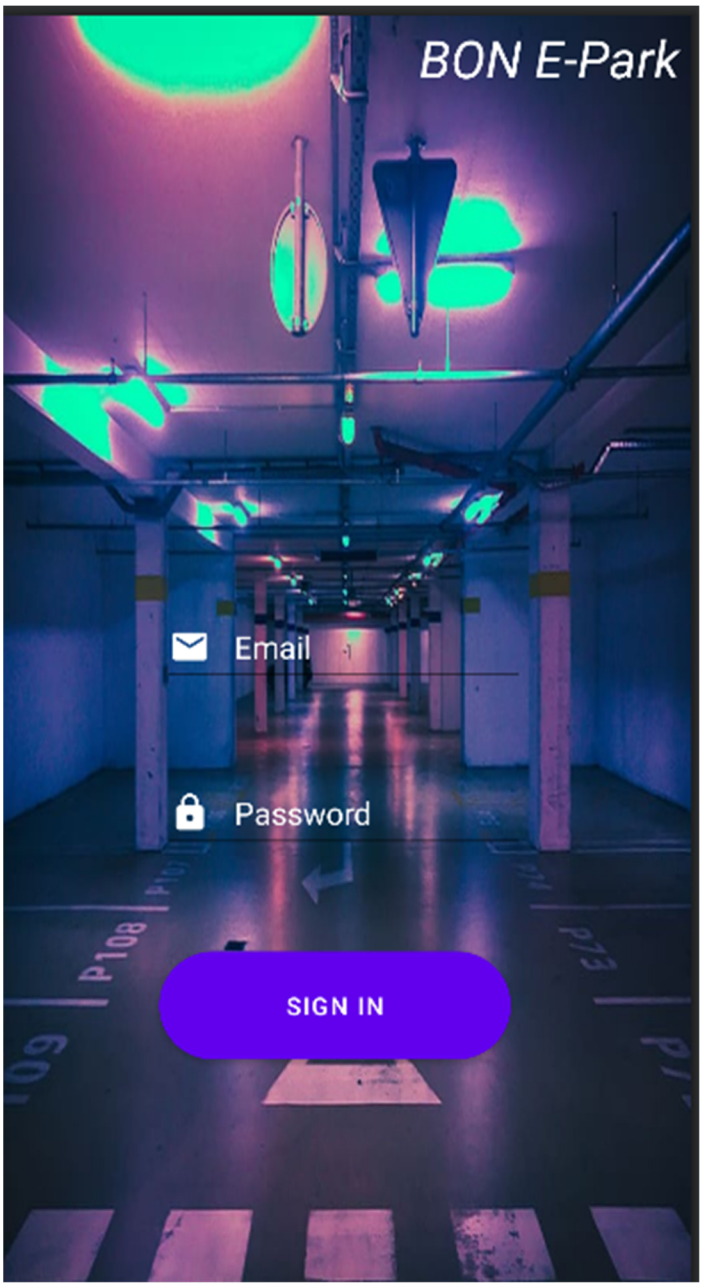
4.2.3 Mobile Phone Application

After validating the user's log-in coordinates, the application starts handling signals using on side thread that handles the signals as an input stream, while receiving, using another thread the application converts the input stream to a readable string, which can be used to know the parking availability.

The application indicates the availability as a car image that keeps appearing and disappearing based on the availability status.

4.3 Mobile Application

4.3.1 Log in Page



4.3.2 Availability page

4.3.2.1 Hide Car (Available)

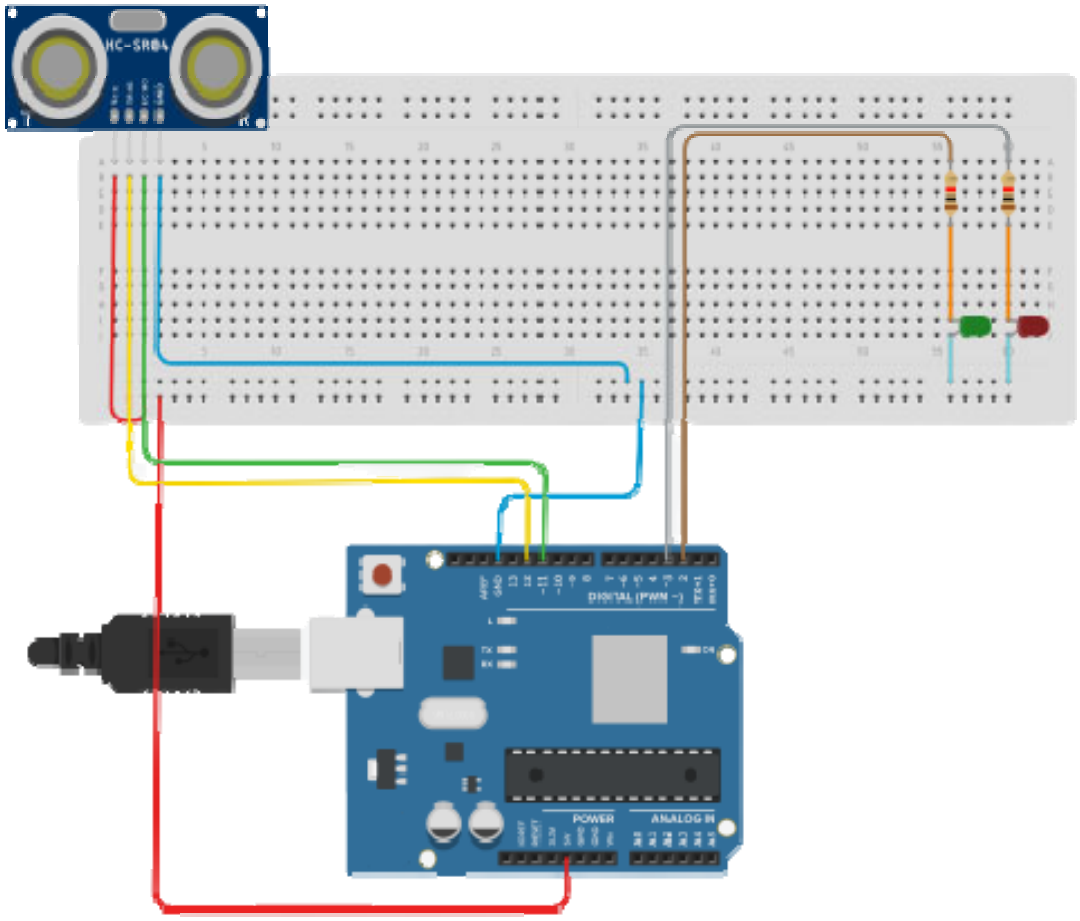
4.3.2.2 Show Car (Occupied)

Chapter Five:

Design

5.1 Hardware Design

5.1.1 Circuit Design (ESP 8266)

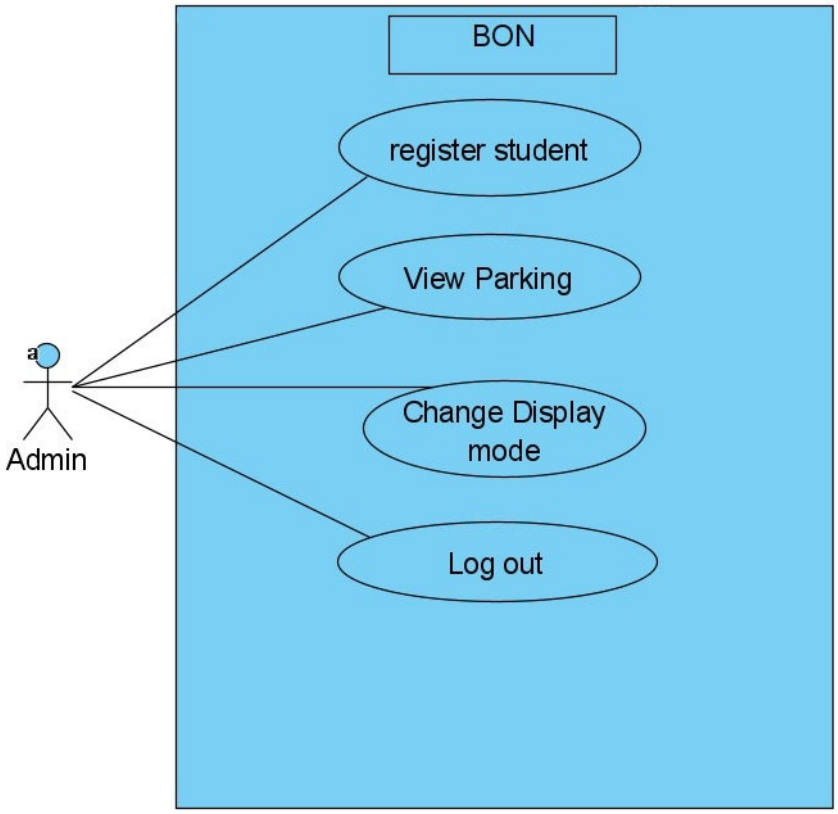


5.2 Software Design

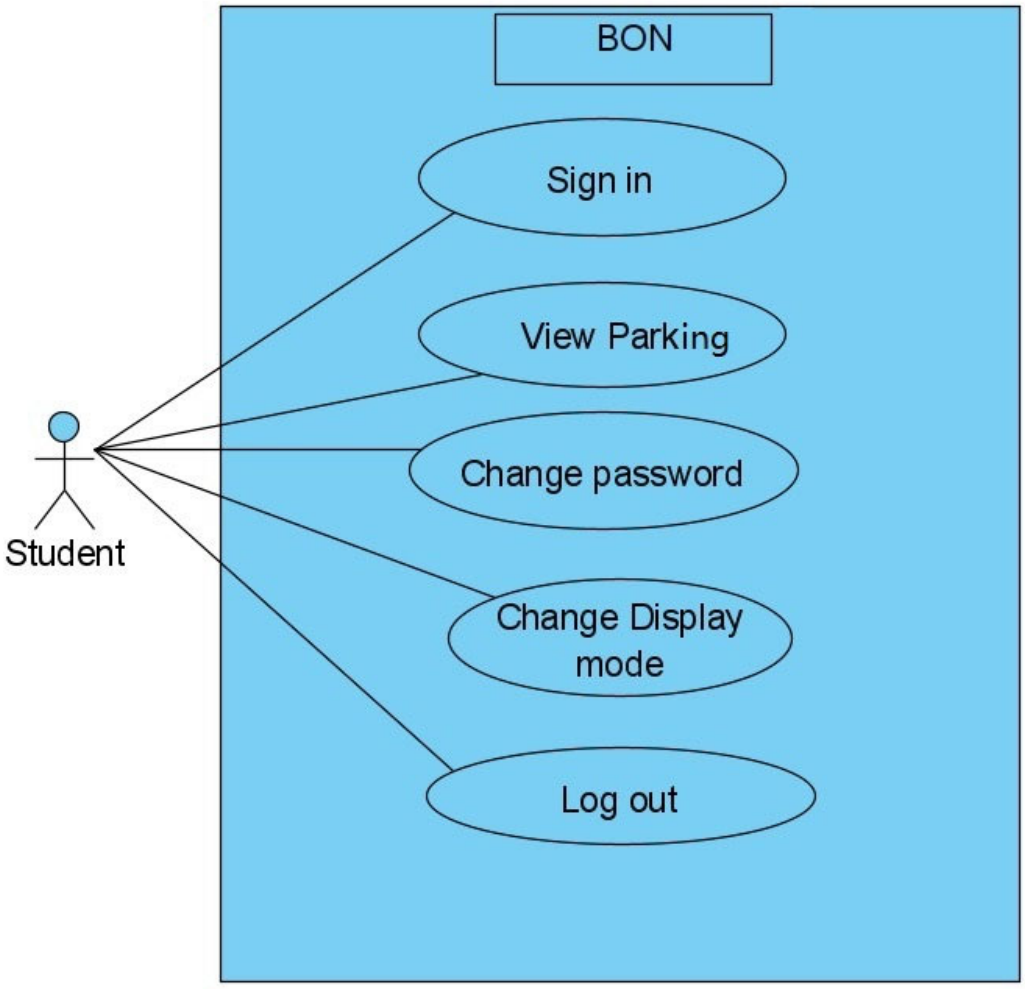
5.2.1 Use case:

Pre-Condition: Sign in

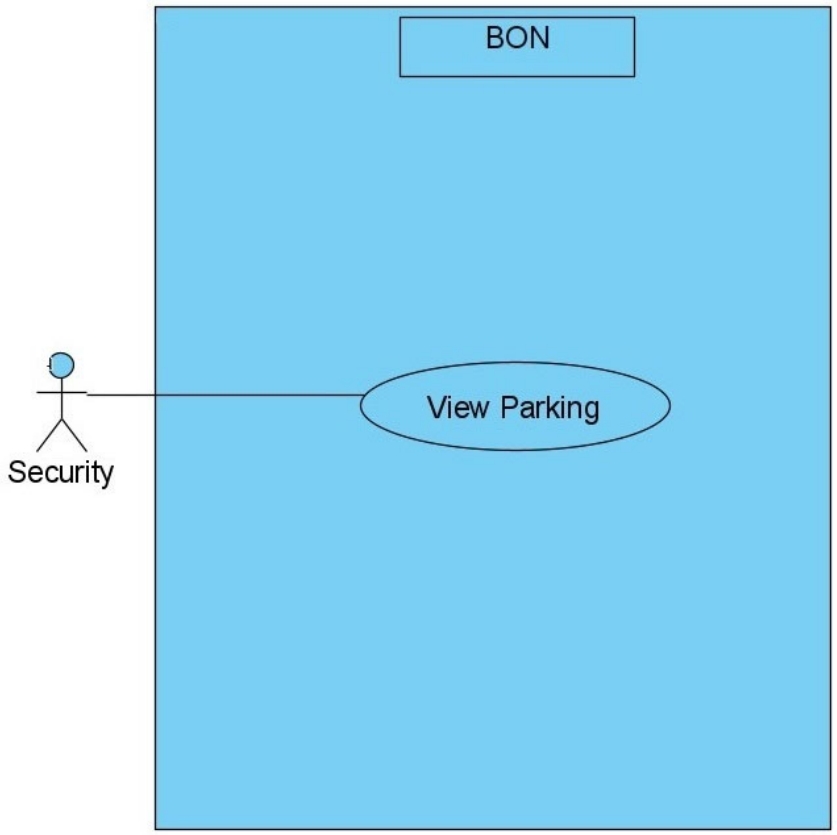
- **Admin**



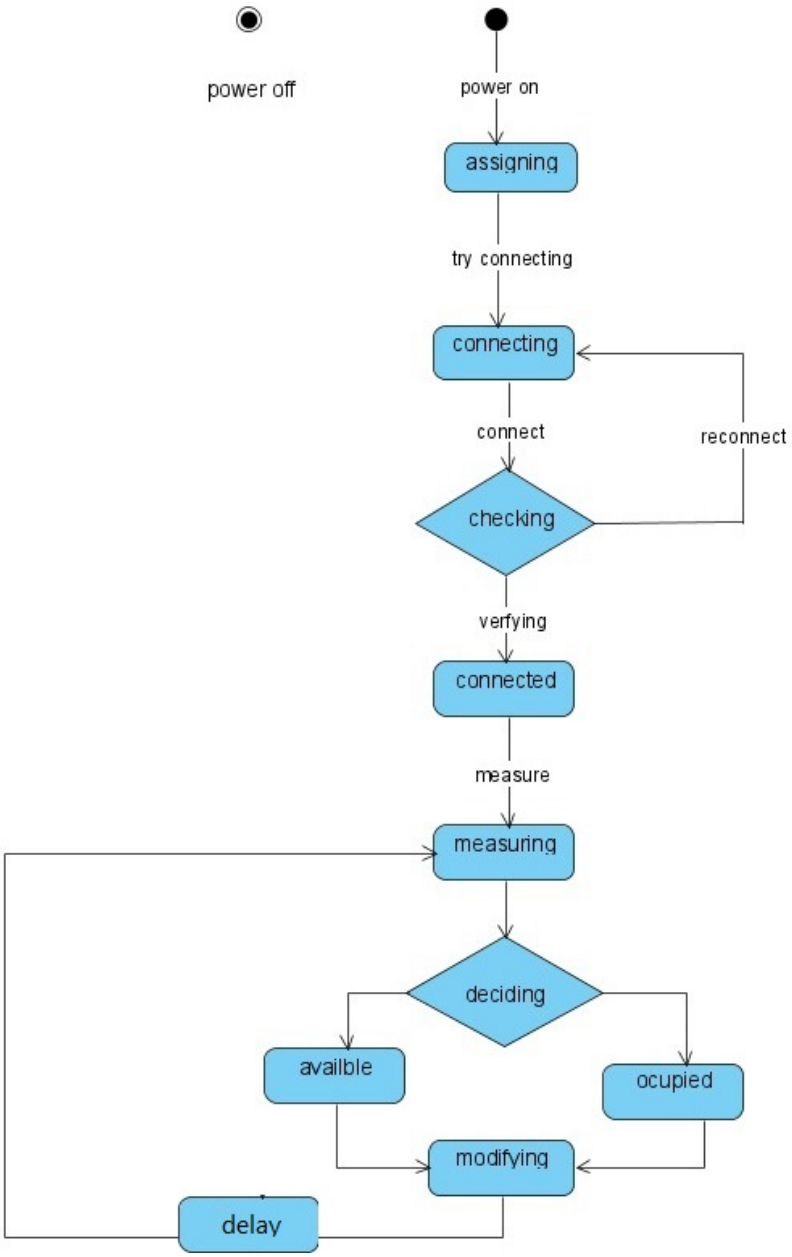
- **Student**


BON
Sign in
View Parking
Change password
Change Display mode
Log out
Student

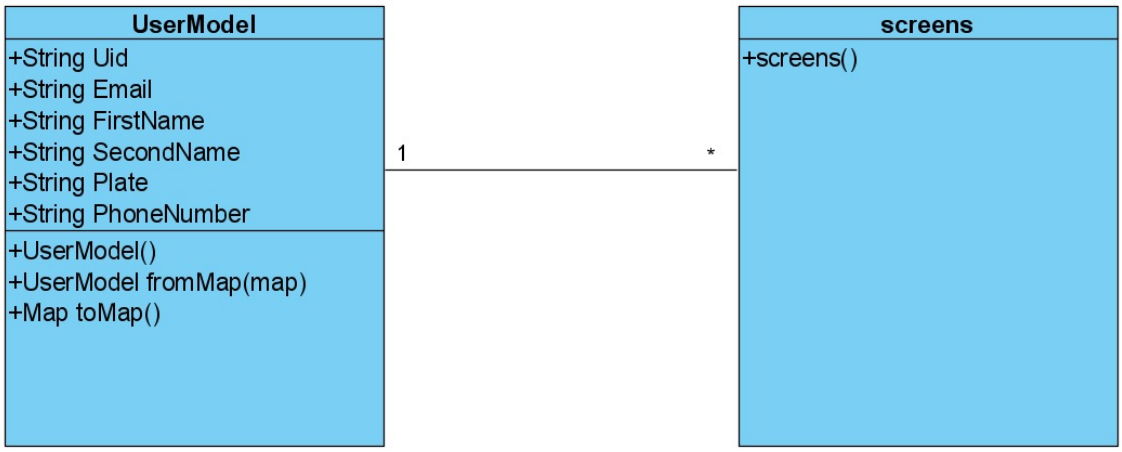
- **Security**



5.2.2 State machine diagram



5.2.3 Class Diagram



Chapter Six:

Software Implementation

6.1 Introduction

This chapter will discuss our project's implementation phase, its main functionalities, and how it performs and works.

6.2 Technology & tools

Name	Description
Android Studio	Android studio is an Integrated development environment used for building mobile phones application.
Arduino IDE	Arduino Integrated development is the implementation code used to write and upload the code to the Arduino device.
Tinker cad	Tinkercad is a design tool used to design and draw Arduino wires and wire connections.
Visual Paradigm	Visual Paradigm is a design tool used to design our project software.
Visual Studio Code	VS Code is a code editor tool that has been used to program vertical prototype python parts.
Google forms	Google form is an online tool to design forms, used to prove problem existence.
Microsoft word	Documentation tool.
Grammarly	Grammar phrasing tool used for documentation section.
Firebase	Firebase is a No SQL online database & real-time database designed by Google, used as the main database for our application.
Flutter	Flutter is a mobile application implementation language from google, flutter is our application's main development language.
C++	C++ is a programing language that has been used to implement Arduino functionality.
Kotlin	Kotlin is an android mobile application implementation language that has been used to implement our vertical prototype functionality.
Python (sockets)	Python is a high-level language, used to implement socket prototype

6.3 Implementation details

6.3.1 Arduino

6.3.1.1 Distance measuring

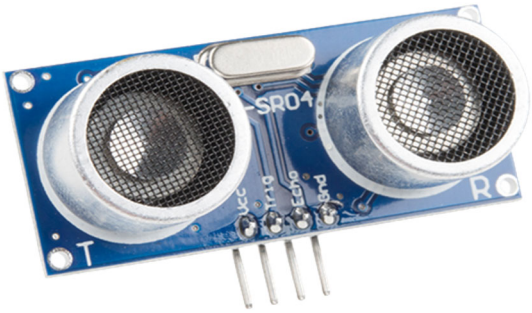
Code sample

```
const int trigPin = 2;
const int echoPin = 15;
int duration = 0;

int distance = 0;

void setup()
{
pinMode(trigPin , OUTPUT);
 pinMode(echoPin , INPUT);
}
void loop()
{
digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration/58.2;
```

Explanation



After assigning the trigger and echo port to their connected position at the Arduino device

In the setup section we determine the output and input port, moreover in the loop section measuring functionality implemented, firstly we make sure that the trigger is already off then it sends the trigger signal and turn it off, after sending the trigger signal echo will stay ready to receive the trigger signal after it returning after hitting item.

The code will measure the time taken to complete this process to get the distance according to the distance measuring formula.

6.3.1.2 Light indicator

Code Sample

```
const int LED1 = 22;
const int LED2 = 23;
void setup()
{
pinMode(LED1 , OUTPUT);
  pinMode(LED2 , OUTPUT);
}
void loop()
{
//After distance measuring
if ( distance < 50 )
  {
digitalWrite(LED1, HIGH);
else
  {
    digitalWrite(LED1, LOW);
  }
if ( distance >= 50 )
  {
    digitalWrite(LED2, HIGH);
}
  else
  {
    digitalWrite(LED2, LOW);}
```

Explanation

Led light indicator used to indicate the result of measuring whether the parking is available or occupied before sending the result to firebase if the result is available to green light will be turned on, In case the result is occupied red light will be turned on.

6.3.1.3 Firebase

Code Sample

```
#include <FirebaseESP32.h>
#define FIREBASE_HOST "https://parking-3d8e1-default-rtdb.europe-west1.firebasedatabase.app/"
#define FIREBASE_AUTH "krcgo6nCyLdC4iPMBGi5zIuaWXQqP11NmsetWnnh"
const char* ssid = "Zhone_F758";
const char* password =  "znid310942296";
FirebaseData firebaseData;
FirebaseJson json;
void setup()
{
WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("...");
  }
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
  Firebase.setReadTimeout(firebaseData, 1000 * 60);
  Firebase.setwriteSizeLimit(firebaseData, "tiny");
}
void loop()
{
if ( distance < 50 )
  {
    digitalWrite(LED1, HIGH);
    json.set("/data", "Ocu");
  Firebase.updateNode(firebaseData,"/Sensor",json);
    delay(100);
  }
  if ( distance >= 50 )
  {
    digitalWrite(LED2, HIGH);
       json.set("/data", "Avi");
  Firebase.updateNode(firebaseData,"/Sensor",json);
    delay(100);
  }
    delay(2000); }
```

Explanation

After connecting to the Wi-Fi network established a firebase connection established between the firebase and the Arduino in the setup phase, moreover in the loop section after measuring the distance the Arduino will modify firebase data with the latest measuring result.

6.3.2 Android Application

Parking screen

Code Sample

```
Timer? timer;
final _database = FirebaseDatabase(databaseURL:"https://parking-3d8e1-default-rtdb.europe-
west1.firebasedatabase.app").reference();
int parkTime=0;
Color Cols=Colors.grey;
@override void initState() {
  _activateListeners();
  timer = Timer.periodic(Duration(seconds: 1), (Timer t) => checkDevice());
    super.initState();
  }
 void checkDevice() {
    setState(() {
      if (((((DateTime
          .now()
          .millisecondsSinceEpoch) / 1000) + 10800) +1 - parkTime) > 10) {
        Cols = Colors.grey;
      }
    });
  }
void _activateListeners() {
    _database
        .child("Sensor/seconds")
        .onValue
        .listen((event) {
      final int sec = event.snapshot.value as int;
      parkTime=sec;
      _database.child("Sensor/data").onValue.listen((events) {
       final String data =events.snapshot.value.toString();
      setState(() {
      if (data == "Avi") {
        Cols = Colors.green;
      } else {
        Cols = Colors.red;
      }
      });
  });
    });
  }
```

Explanation

Parking Screen mainly contains two functions:-

activeListner ( ): active listener is a void function that contains an event listener that listens only when firebase data changes it listens on time and availability status changing

checkDevice ( ): check device is a void function run repeatedly by a timer, it compares current Epoch time with time that came from firebase to clarify if the device is working or not.

6.4 Libraries

6.4.1 Arduino

Library	Description
#include <WiFi.h>	This library enables our Arduino to use its Wi-Fi chipset
#include <FirebaseESP32.h>	This library enables our Arduino to use firebase's real-time database service and update the desired field.
#include <NTPClient.h>	This library enables our Arduino to use and create time objects.
#include <WiFiUdp.h>	This library is required to enable Arduino to use the time object.

6.4.2 Flutter

Library	Description
Animated_splash_screen.dart	Splash screen
Firebase_core.dart	App initialization
Firebase_auth.dart	Create authentication between application and firebase coordinates
Firebase_database.dart	Enable using firebase data
Fluttertoast.dart	Notify user (toast message)

Chapter Seven:

Test Document

7.1 Introduction

This chapter presents the test plan document and phase for our system BON E-Parking Application against its software requirement.

7.2 Test Item

- Admin Account
- Students Account
- Arduino

7.3 Features To Be Tested

Admin Account

1. Sign in
2. Register Student
3. See parking

Student Account

1. Sign in
2. Sign out
3. Forget Password
4. Change Password
5. See Parking

Arduino

1. Parking Availability
2. Status

7.4 Features Not To be Tested

1. Usability
2. Dark mode
3. portability

7.5 Test Methods

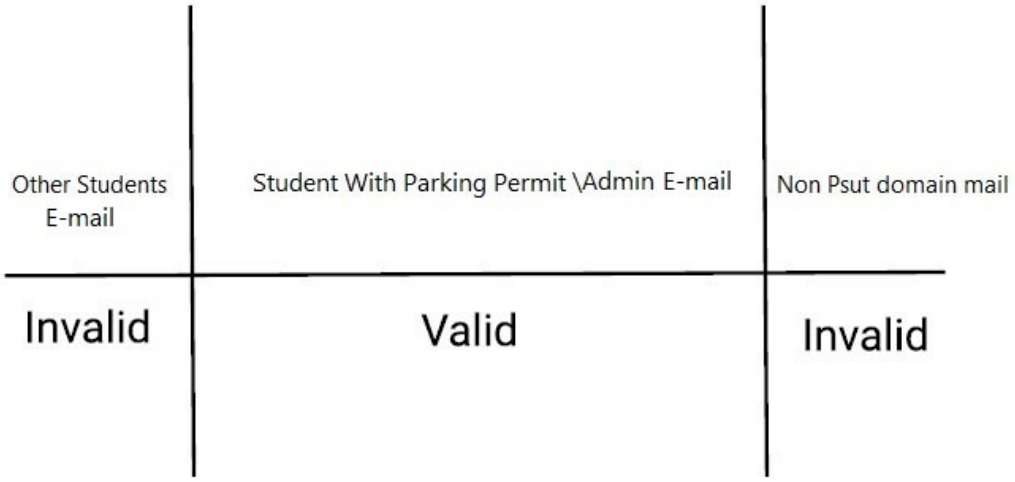
7.5.1 Black-Box Testing

- Equivalence partitions:

Equivalence partitions are used to indicate limits, valid, and invalid partitions in our application.

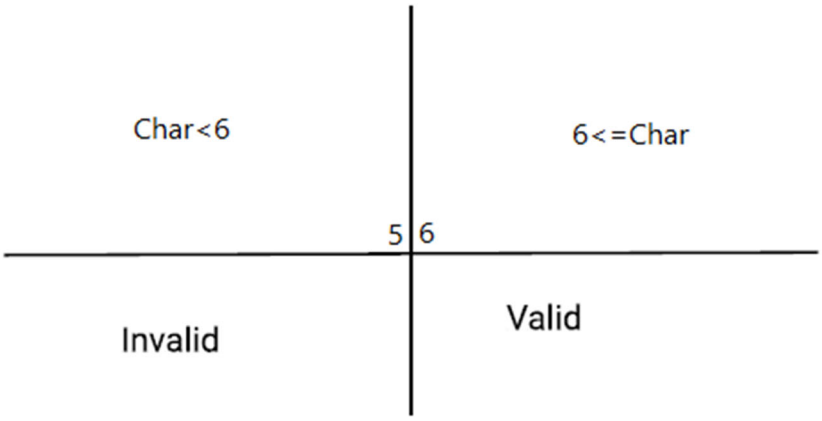
❖ **Log In Page**

Email Field



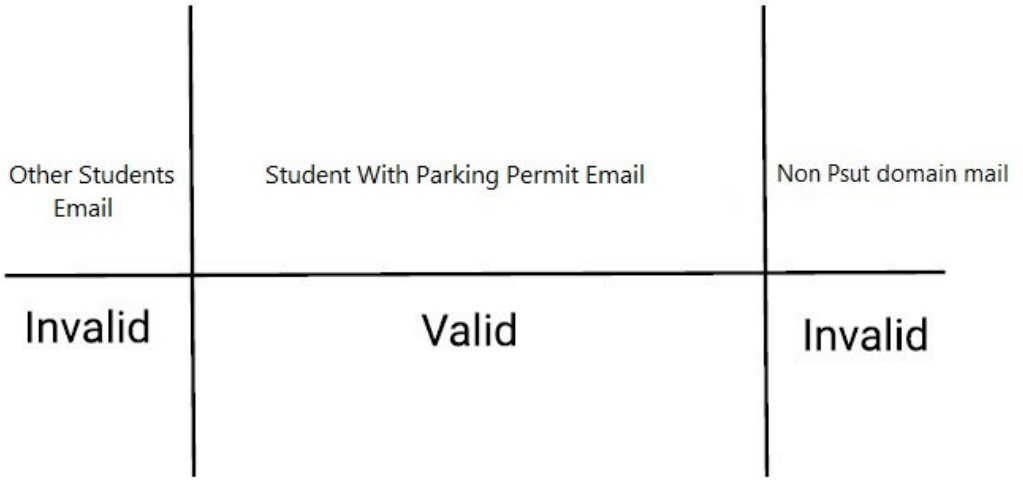
Sign in Email Field

Password Field



Password Field

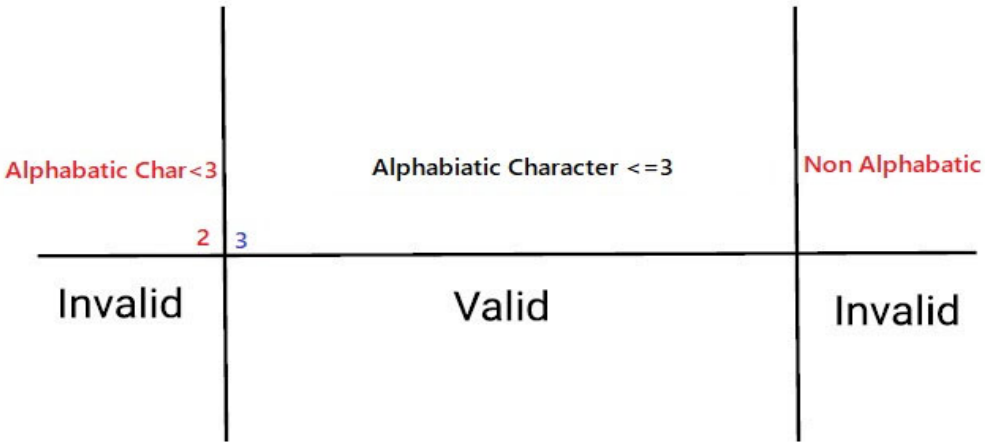
Forget Password Field



Forget Password Email Field

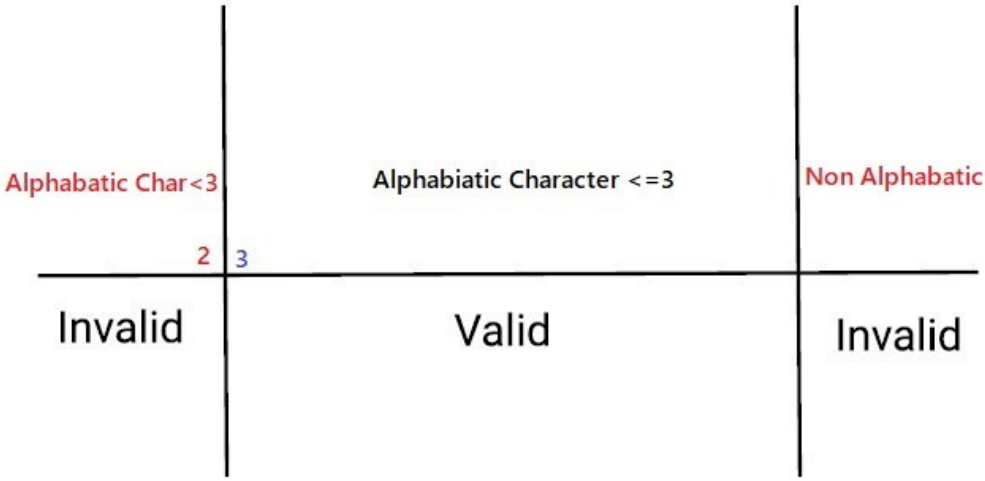
❖ **Admin (Student Registration Screen)**

First Name Field:



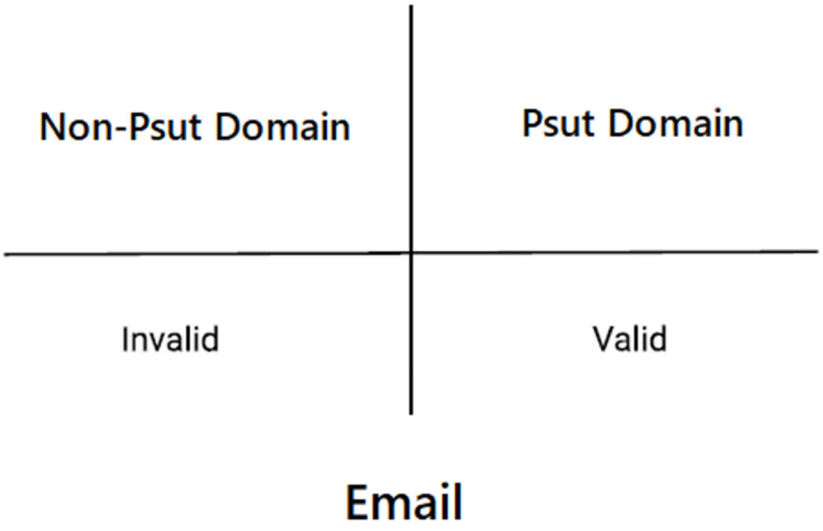
First name Field

Last Name Field:

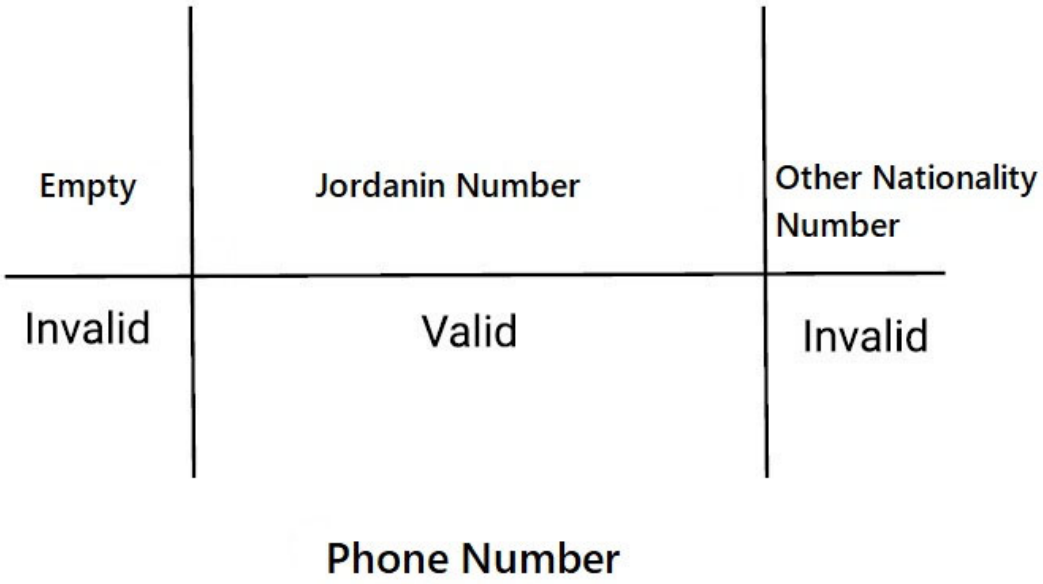


Last name Field

Student Email Field



Car Plate Field



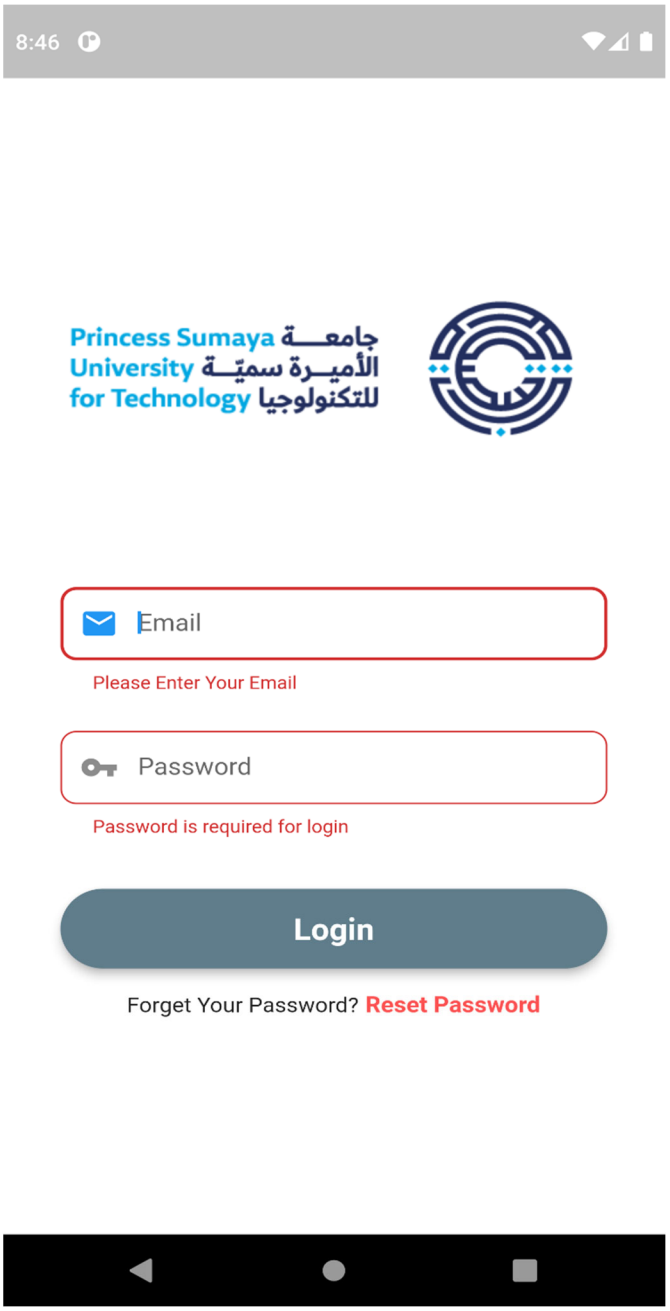
➢ Data Testing

For the data testing technique three methods will be used:

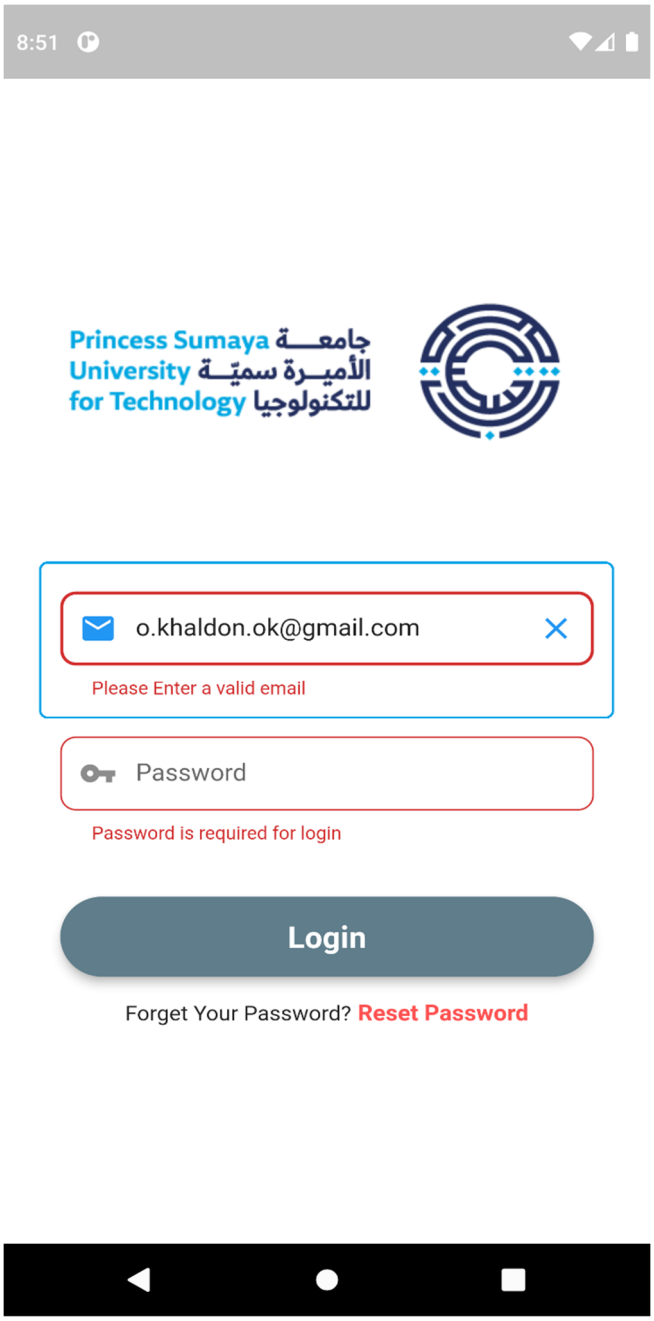
1. Boundary testing.
2. Null testing.
3. Bad data testing.

I. Sign in:

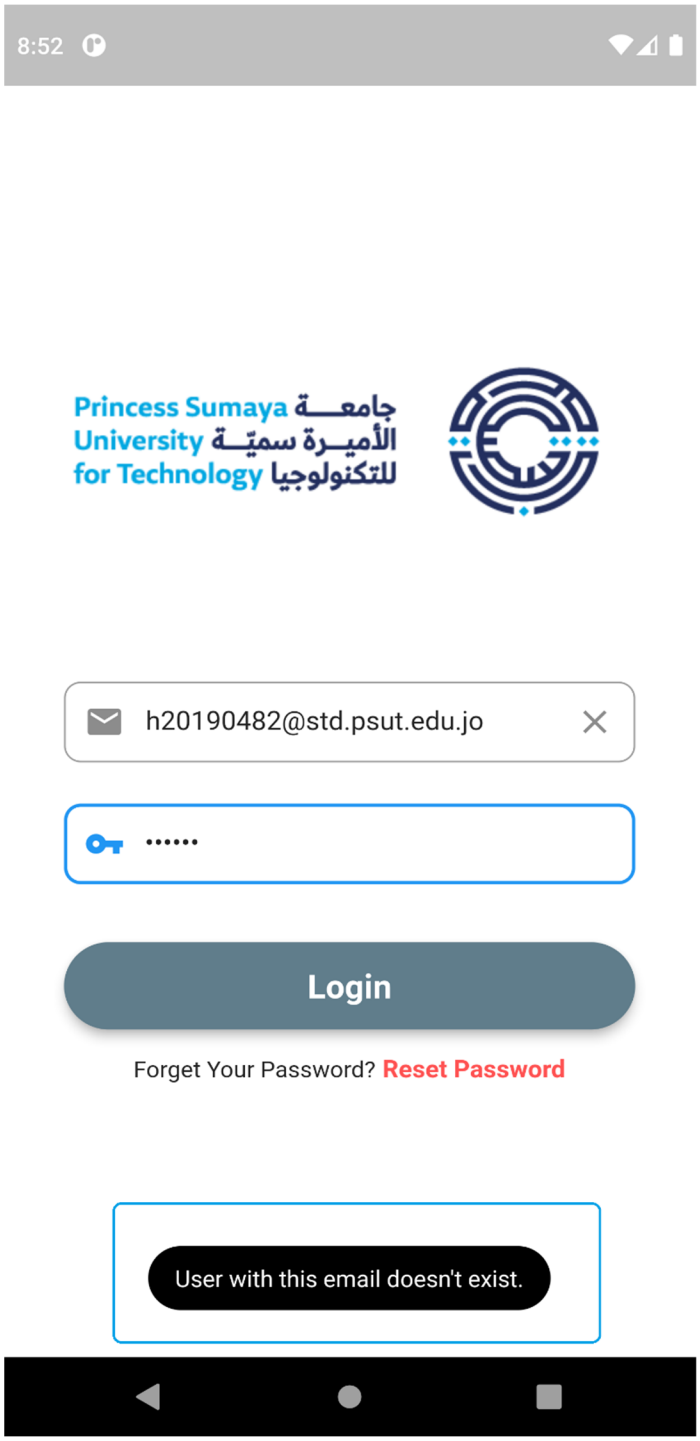
-Null Entry



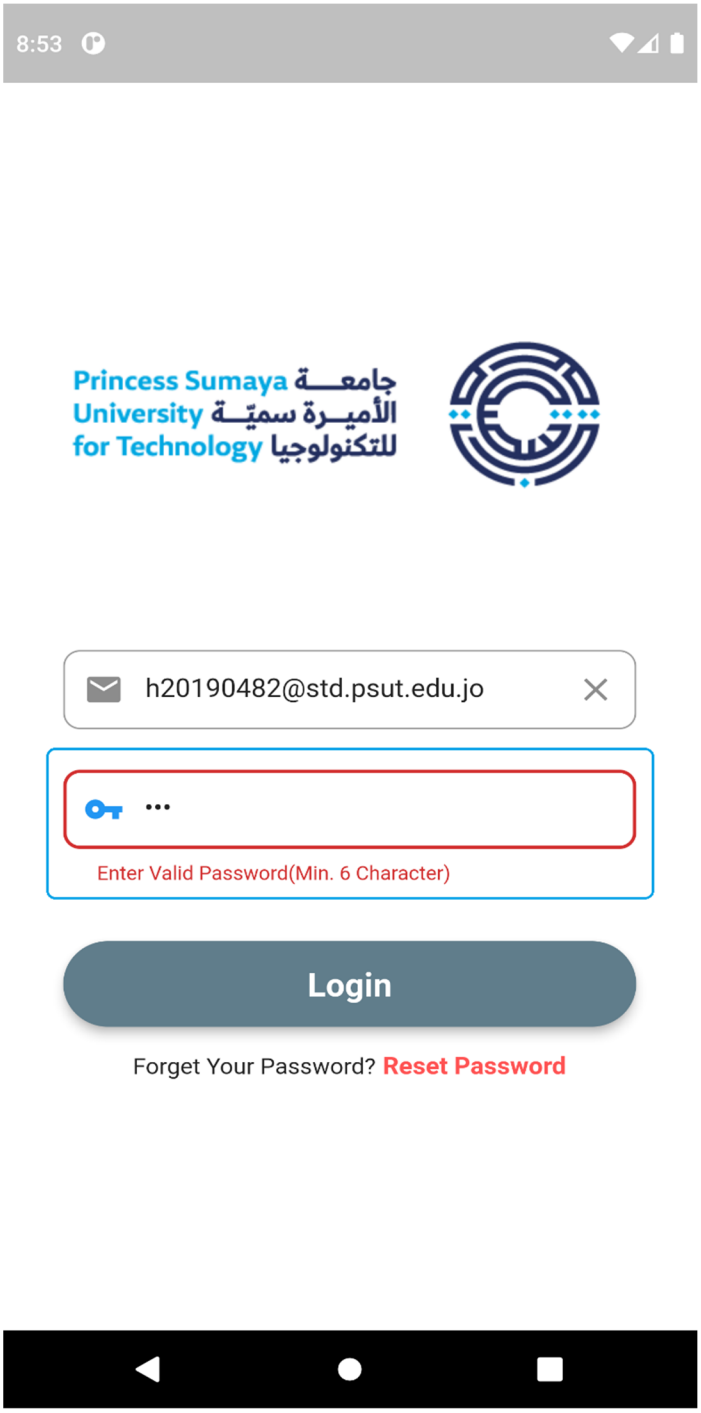
-Non-PSUT Email



-Student with No Car Permit

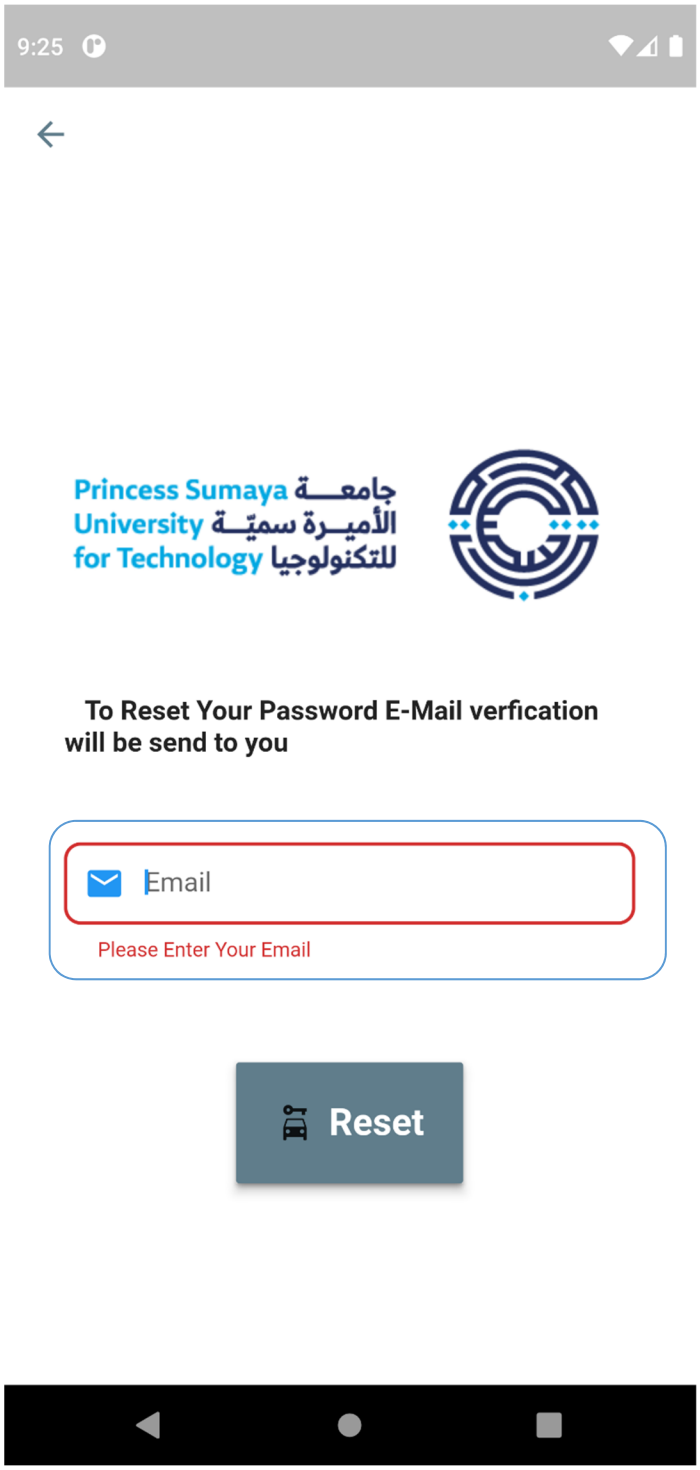


-Password Less than 3 Char:

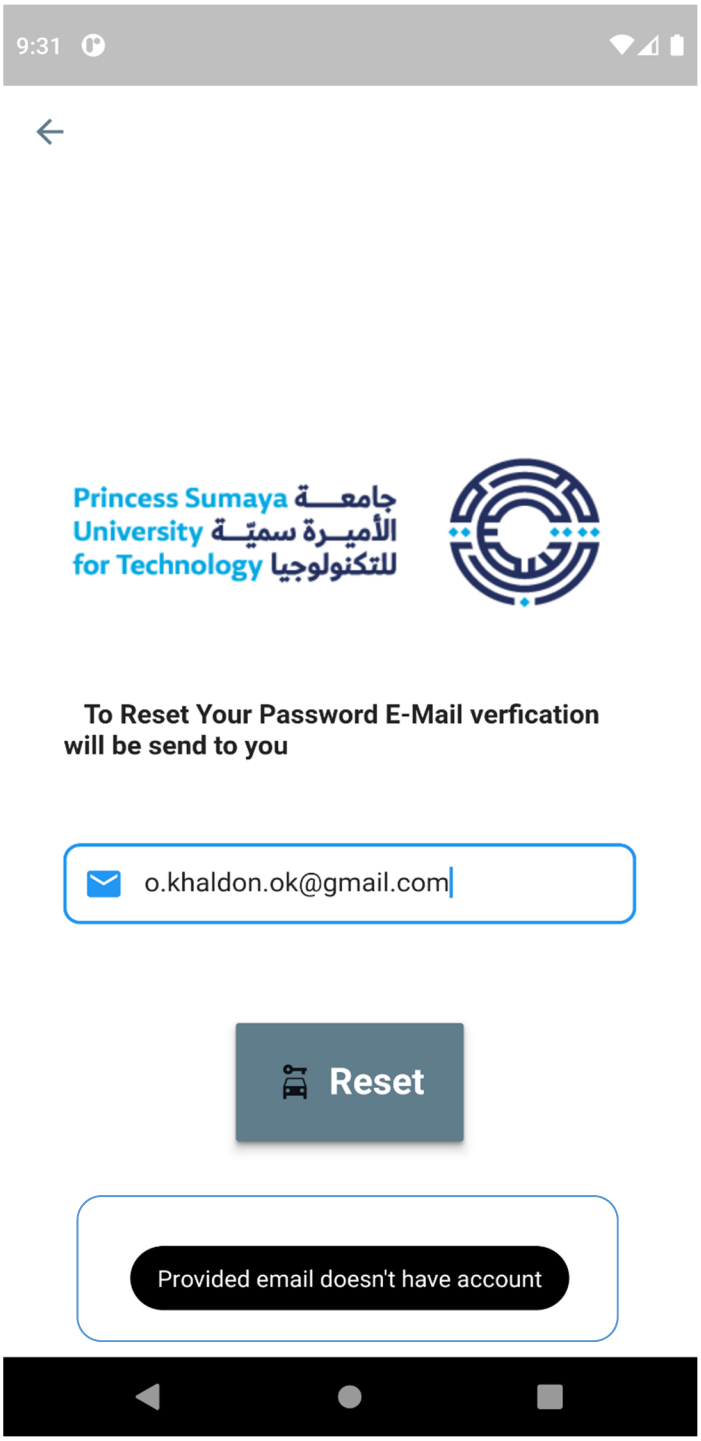


II. Forget password

-Null Entry

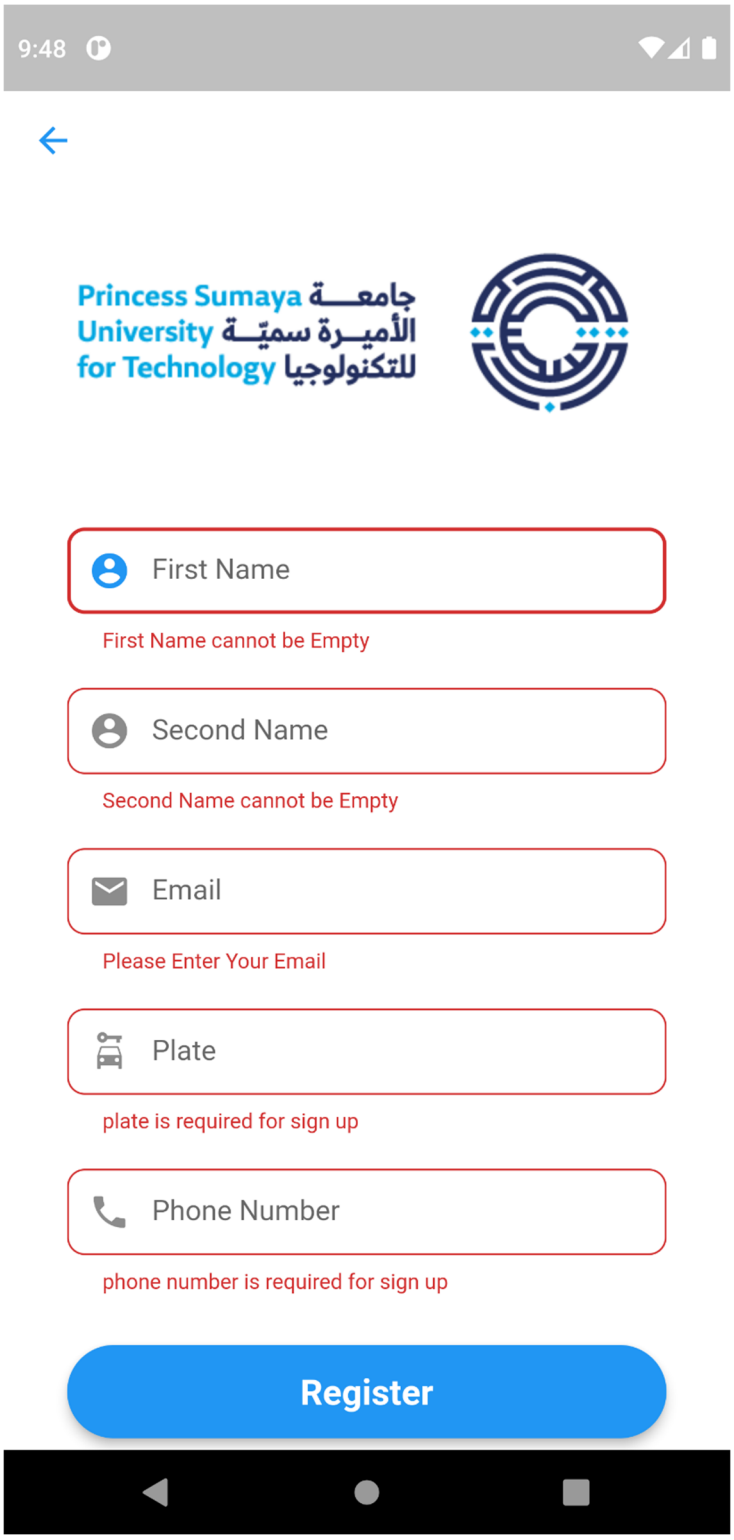


-Email with No Account

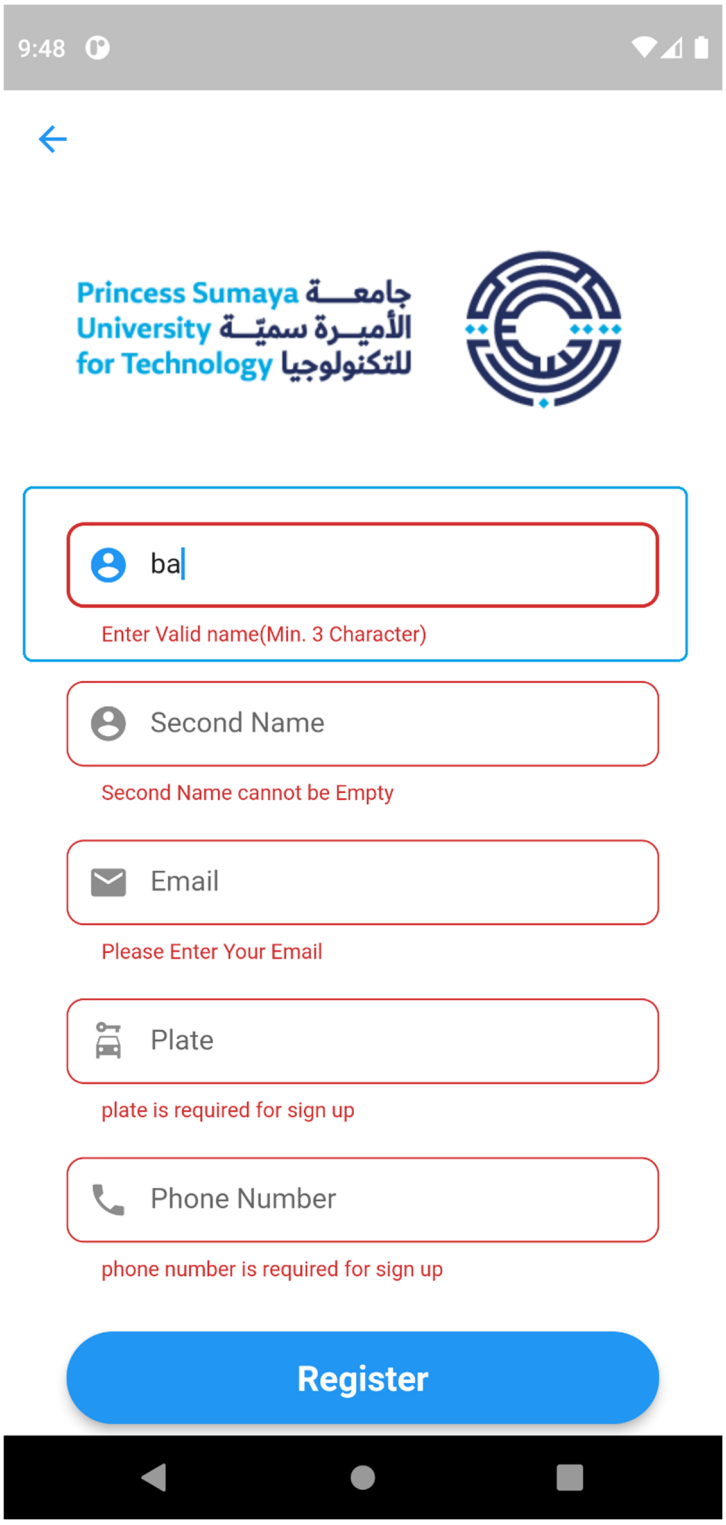


III. Sign up

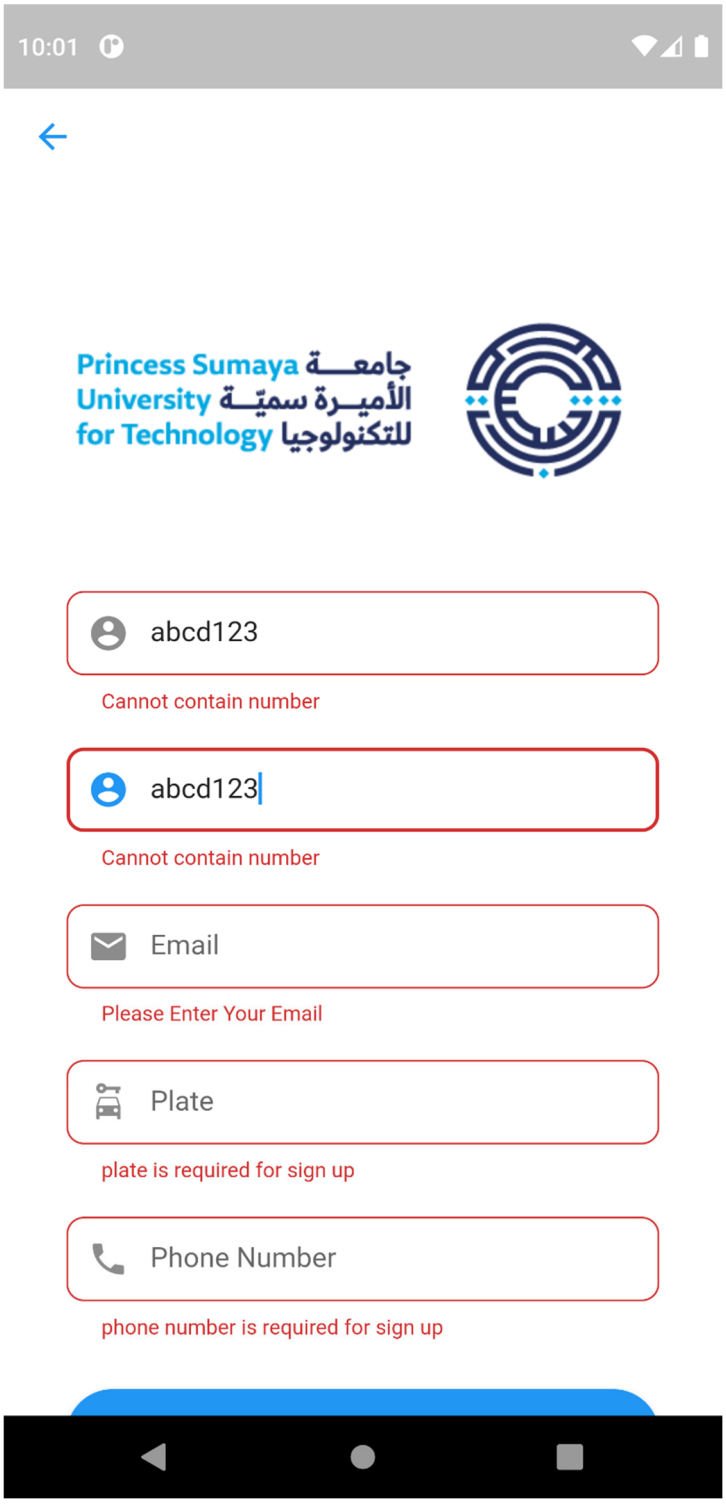
-Null Entry



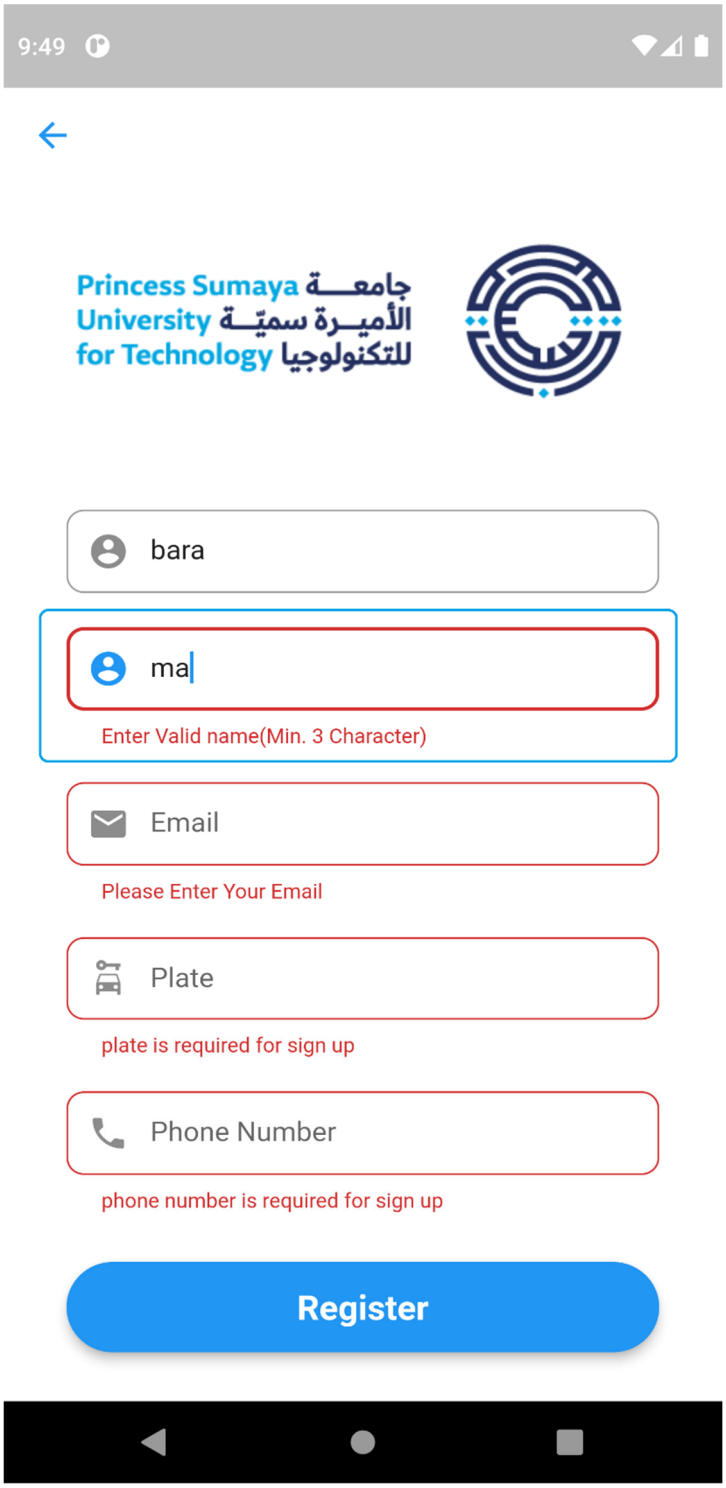
-First Name Less than 3 Char



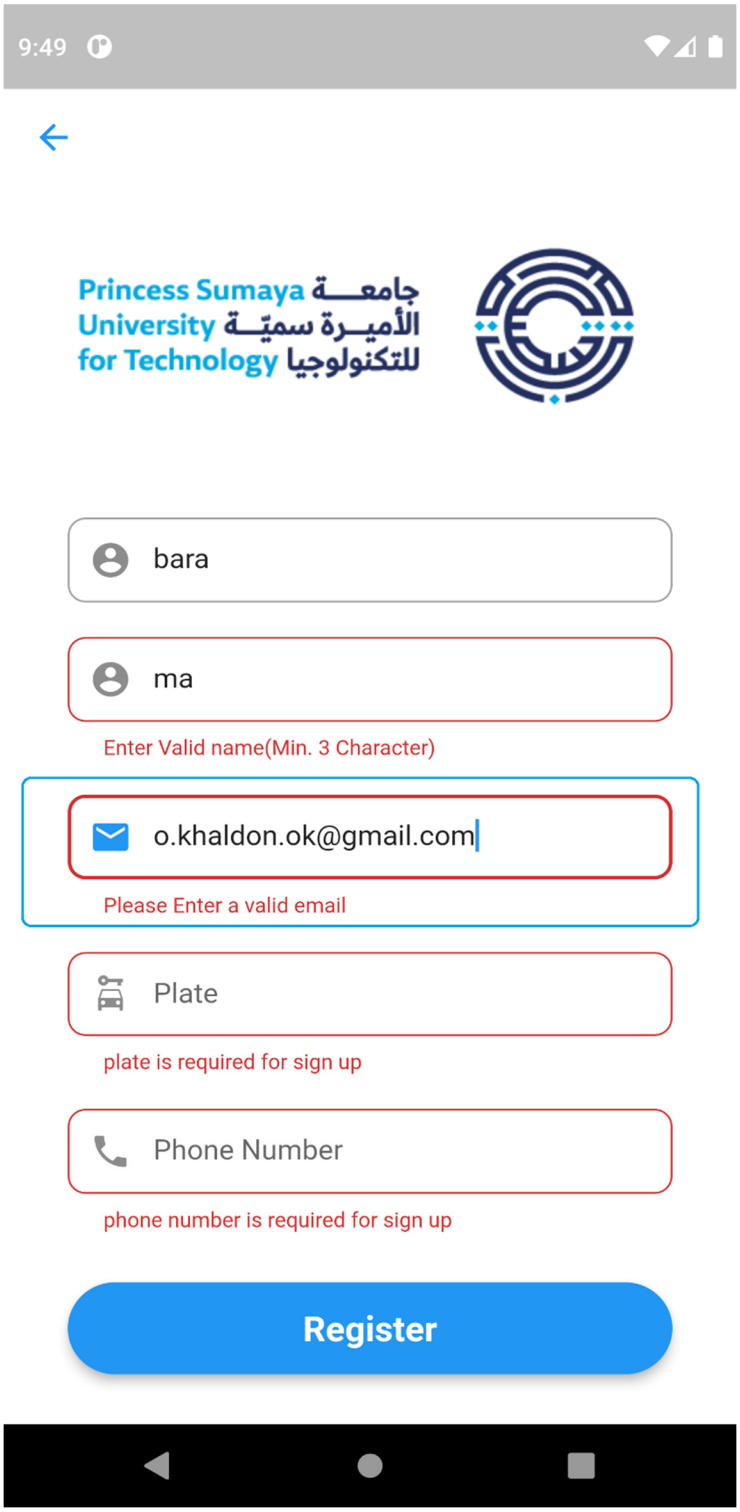
-Name field Contain Number



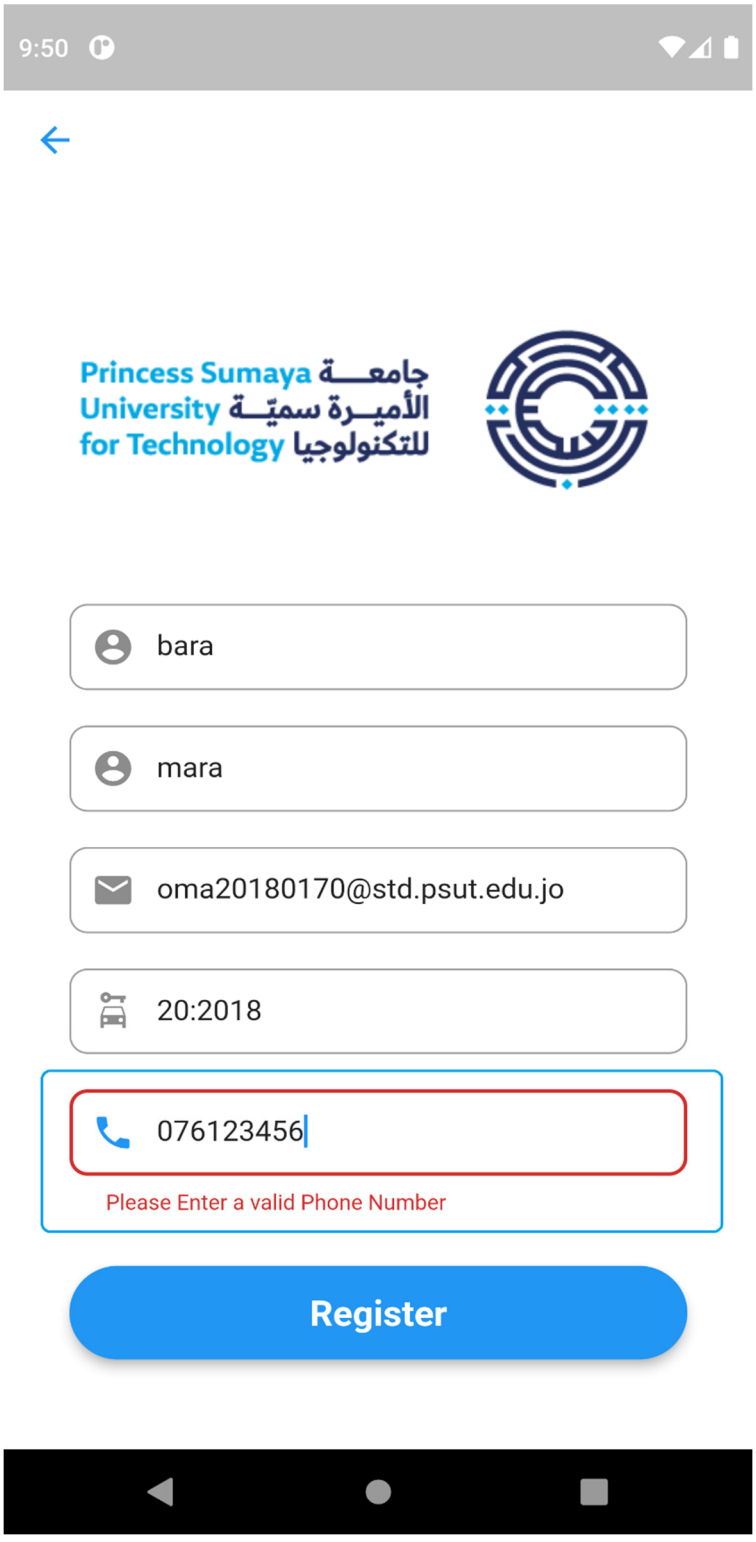
-Last Name Less Than 3 Char



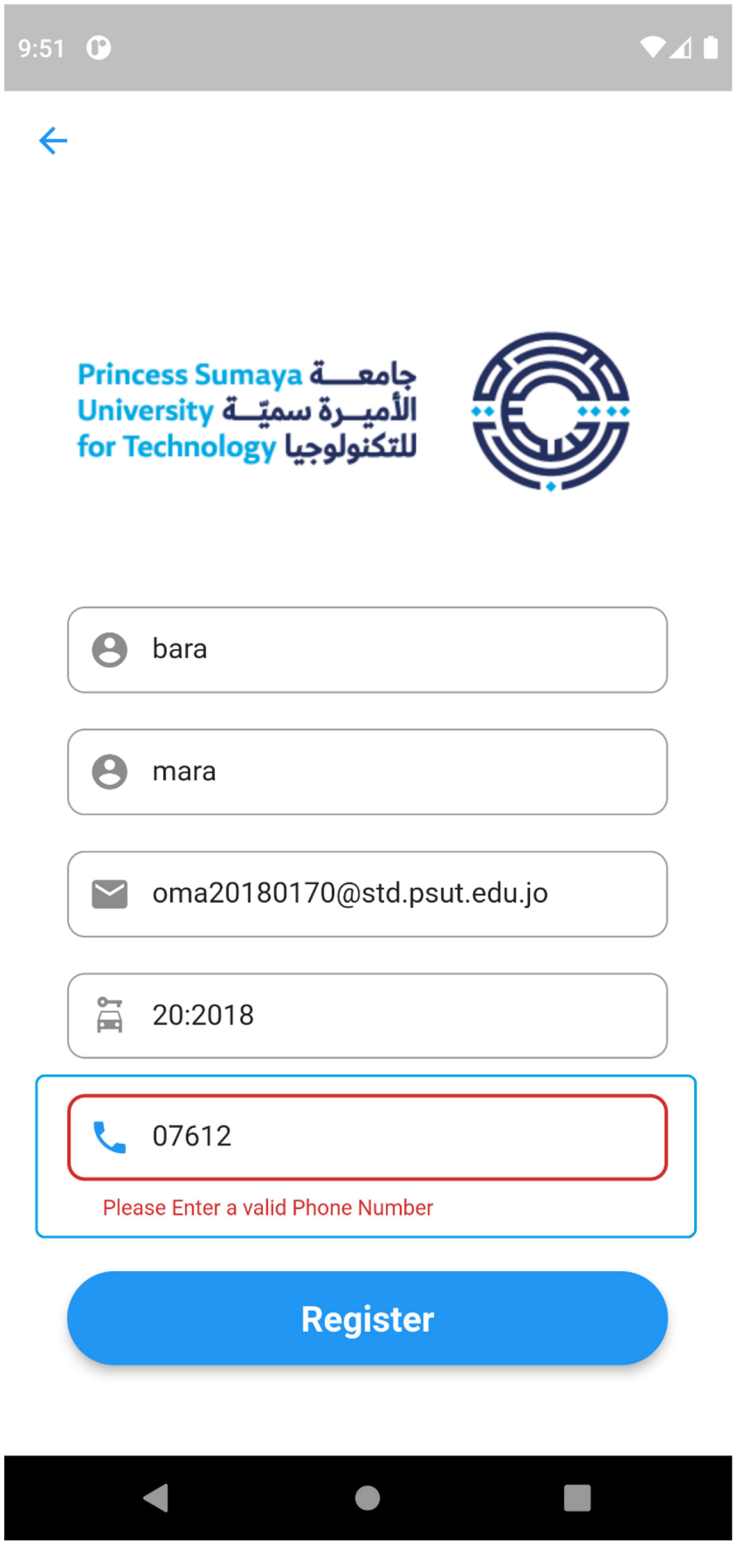
-Non-PSUT Domain Mail



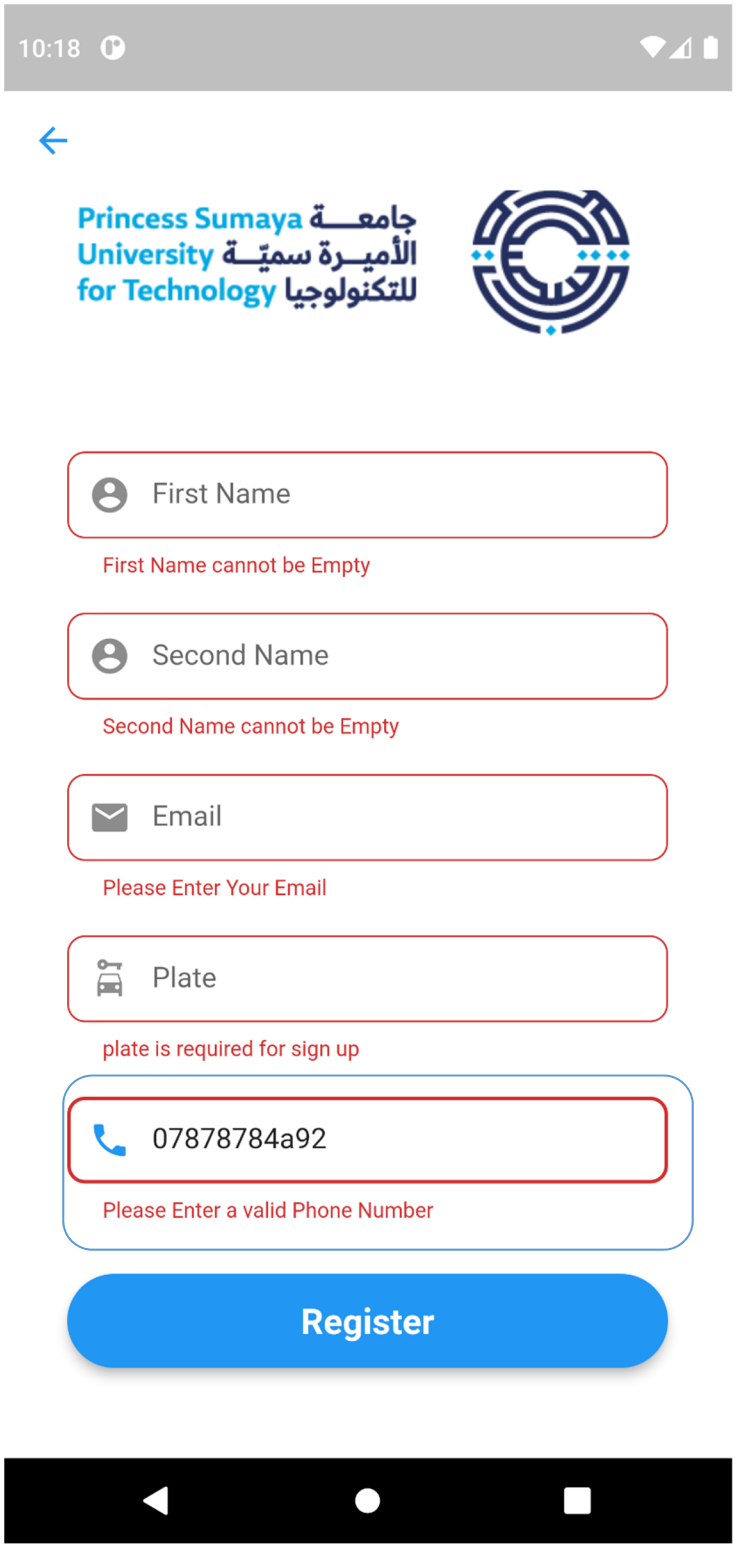
-Non-Jordanian Phone Number



-Not A Phone Number



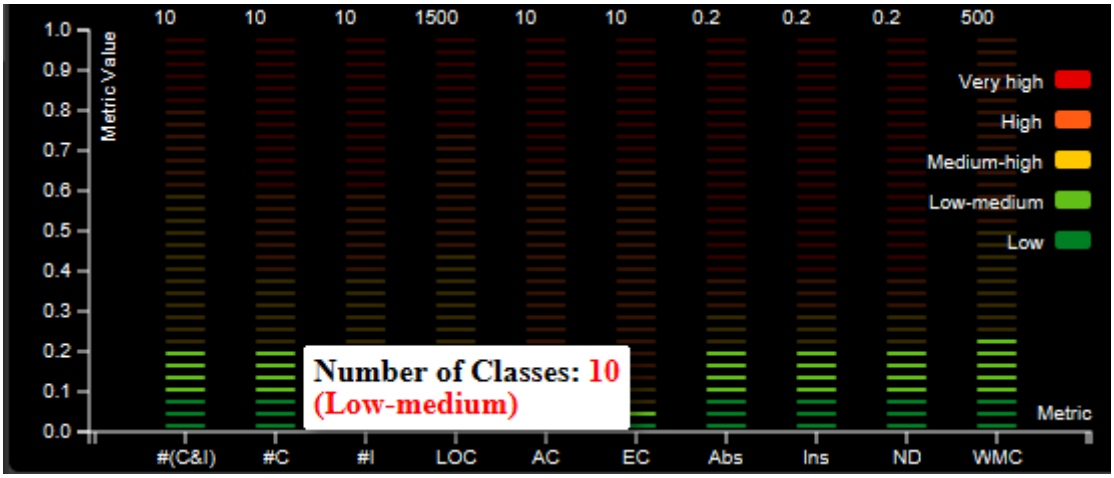
-Contain Letter

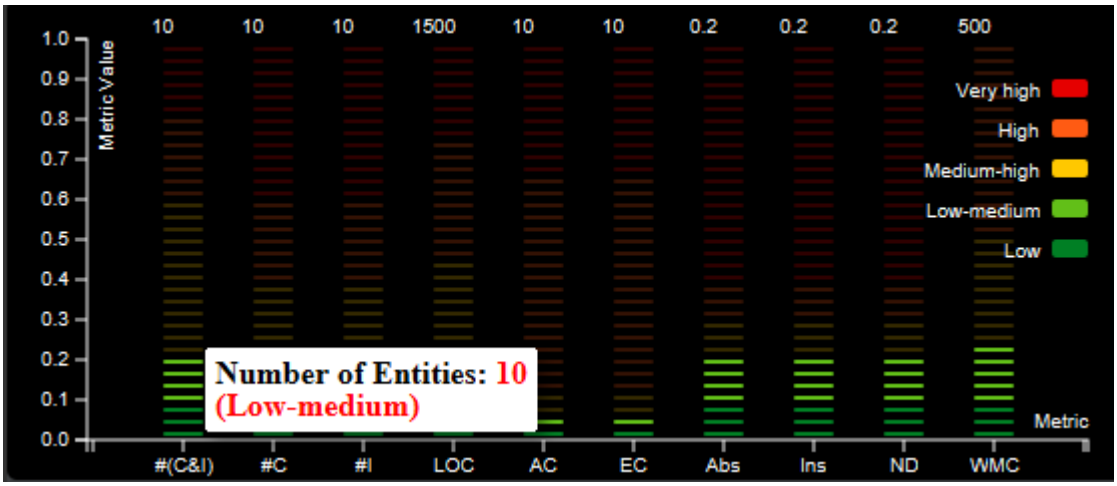


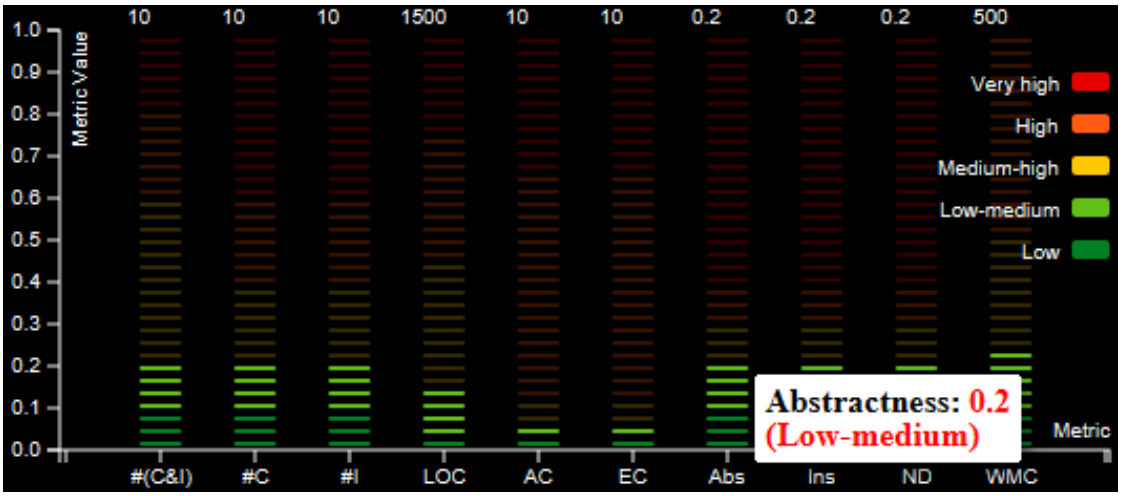
7.5.2 White-Box Testing

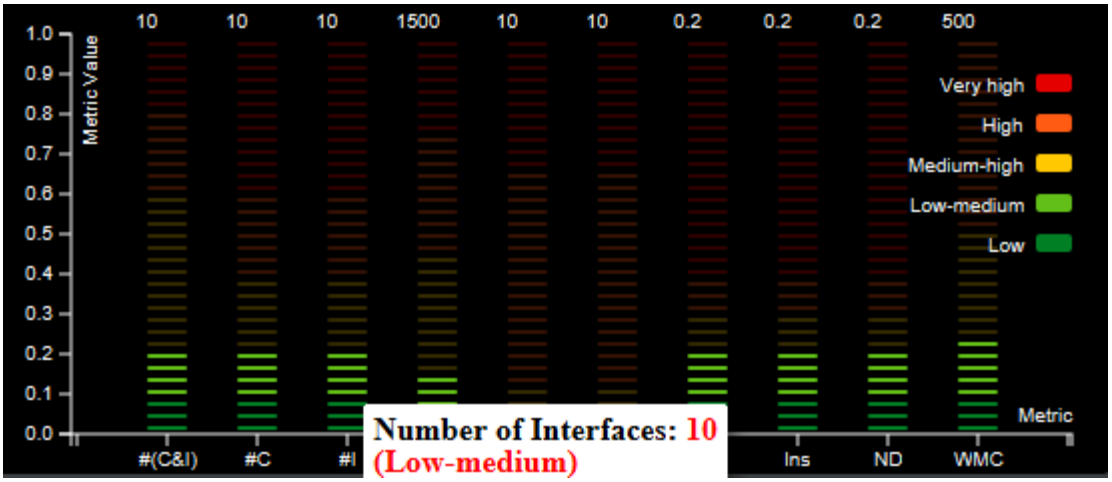
- Static Testing

For this type CodeMR Tool has been used to conduct these results



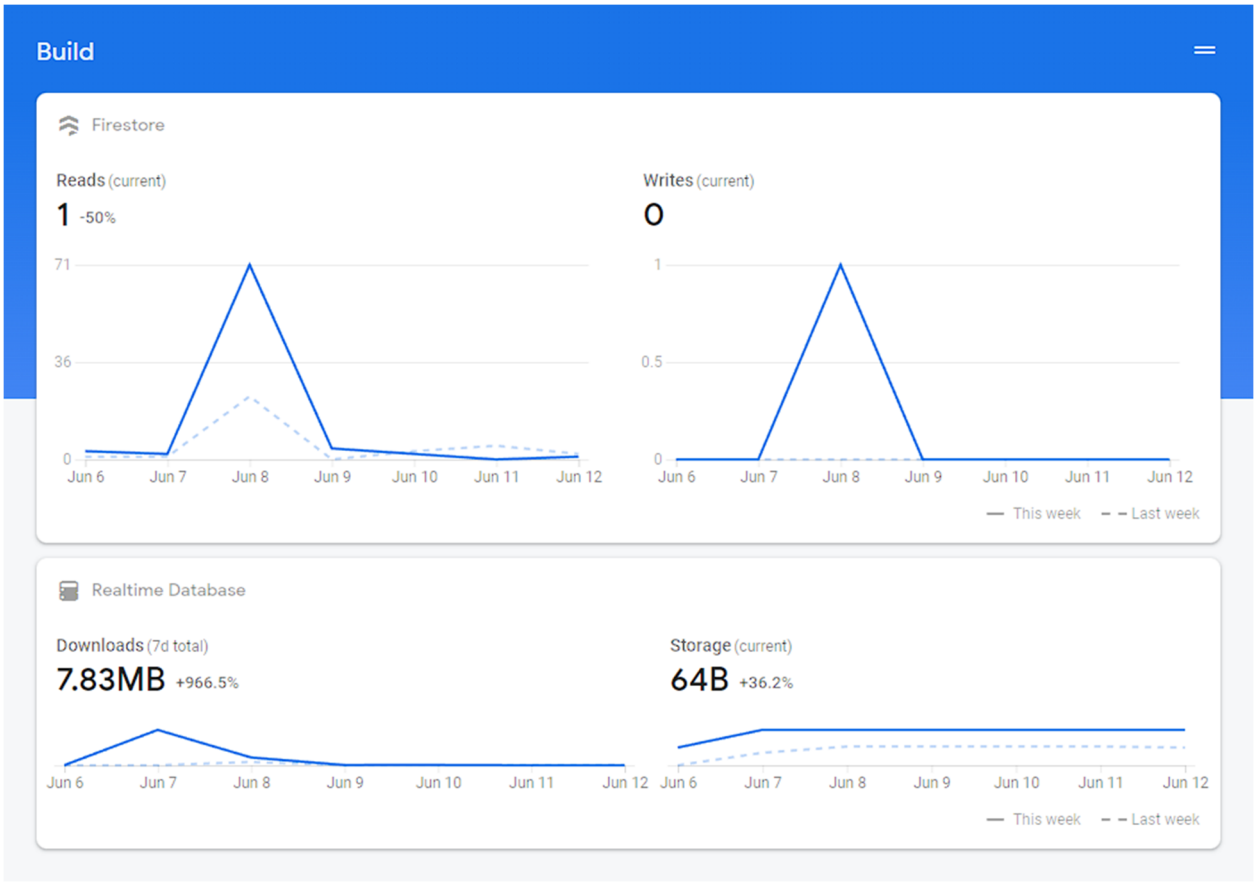






Total Number Line of Codes: 1783 Line

Firebase Load



Chapter Eight:

Future Works

8.1 Introduction

This chapter presents the Future that can be done on our project to improve its functionality.

8.2 Future Plans

8.2.1 Power Source

Currently, the main power source for our Arduino is a normal battery, although this can be modified, adding a sunlight power source will improve the stability of our device and make it consistent.

8.2.2 Functions

- Implement left-up functionality

Some functionality was left-up due to our project time shortage.

- Student ID verification

Adding an ID section to our application will make our application more usable by the student.

References

[1] P. B. ,. V. Hemant Chaudhary, "Advanced CAR parking system using Arduino," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, india, 2017.

[2] S. A. 2. ,. M. 3. C. K. J. Shruthi Mudaliar 1, "IoT Based Smart Car Parking System," researchgate.net, India, 2019.

[3] H. Chaudhary, P. Bansal and B. Valarmathi, "Advanced CAR parking system using Arduino," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2017.

[4] N. M. P. L. a. P. K. Amin Kianpisheh, "Smart Parking System (SPS) Architecture Using Ultrasonic Detector," *International Journal of Software Engineering and Its Applications,* vol. 6, 2012.

[5] H. Lee, D. Kang and W. Moon, "Design and Fabrication of the High Directional Ultrasonic Ranging Sensor to Enhance the Spatial Resolution," in *TRANSDUCERS 2007 - 2007 International Solid-State Sensors, Actuators and Microsystems Conference*, Lyon, France, 2007.

[6] S. Thakare and P. Bhagat, "Arduino-Based Smart Irrigation Using Sensors and ESP8266 WiFi Module," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018.

[7] Y. A. Badamasi, "The working principle of an Arduino," in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, 2014.

[8] S. Mendiratta, D. Dey and D. R. Sona, "Automatic car parking system with visual indicator along with IoT," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, Vellore, India, 2017.