In-Station Delivery Service Safety Monitoring

Final Report

Kush Baheti (3035436583)
Pranav Gupta (3035453646)
Siddharth Dev Tiwari (3035436791)

Supervisor: Dr. Dirk Schnieders
Second Examiner: Dr. Ping Luo

Department of Computer Science
Abstract

The Mass Transit Railway (MTR) is a public transport system in Hong Kong. MTR stations have 1492 stores, requiring restocking of goods, which is done by delivery personnel [1]. MTR stations have seen multiple accidents involving passengers and delivery personnel [2].

This technical paper presents the development of an Indoor Positioning and Alert system to track movements of and violations committed by delivery personnel. The system was developed using Radio-Frequency Identification (RFID) technology. Delivery personnel were equipped with RFID readers. Passive RFID tags were used to construct a virtual map of indoor environments. This allowed real-time tracking of position and associated timestamp of delivery personnel. The data was used to calculate occurrence of violations. A web application was developed to provide alerts, data analysis, and visualization. The key focus was on localization and visualization.

As a result, a cost-effective and scalable Indoor Positioning System prototype with non-invasive infrastructure was developed, with utility over a range of applications. Further research can be conducted on tag deployment strategies, reader signal collisions and signal strength algorithms.
Acknowledgements

We would like to express our sincere gratitude to Dr. Dirk Schnieders for his constant support and invaluable inputs during the course of this project. We would also like to extend our thanks to the the Department of Computer Science and the Department of Electrical and Electronic Engineering for providing us with the opportunity to work on a project with the MTR, allowing us to gain exposure of working on large-scale product development with an industrial giant. We would like to thank the MTRCL for their patience generosity in helping us understand the problem statement and tending to our queries. Lastly, we would like to thank Mr. Francis Tai and Mr. Kevin Wan of Hong Kong RFID Limited for their support in ensuring this project is a success.
# Table of Contents

ABSTRACT ....................................................................................................................... 1

ACKNOWLEDGEMENTS .................................................................................................... 3

TABLE OF CONTENTS .................................................................................................... 4

LIST OF FIGURES ............................................................................................................ 6

LIST OF TABLES .............................................................................................................. 8

LIST OF ABBREVIATIONS ................................................................................................. 9

CHAPTER 1: INTRODUCTION ............................................................................................ 10
  1.1 BACKGROUND ............................................................................................................. 10
  1.2 MOTIVATION .............................................................................................................. 11
  1.3 OBJECTIVES ............................................................................................................ 11
  1.4 SCOPE .................................................................................................................... 12
  1.5 OUTLINE ................................................................................................................ 13

CHAPTER 2: LITERATURE REVIEW .................................................................................. 14
  2.1 INDOOR POSITION SYSTEM .................................................................................. 14
  2.2 FAILURE OF GPS IN AN IPS ................................................................................ 18
  2.3 CHOOSING A GOOD IPS ...................................................................................... 20
  2.4 EFFICACY OF RFID ............................................................................................ 21

CHAPTER 3: DESIGN ....................................................................................................... 23
  3.1 SOLUTION ............................................................................................................ 23
  3.2 IMPLEMENTATION ............................................................................................... 25
  3.3 DIGITAL MAP ....................................................................................................... 25
  3.4 MOBILE APPLICATION ......................................................................................... 27
  3.5 BACKEND ............................................................................................................ 33
  3.6 LOCALIZATION .................................................................................................... 35
     3.6.1 Reader and Tag Localization ........................................................................ 36
     3.6.2 Active and Passive Tags .............................................................................. 38
     3.6.3 Tag Frequency .............................................................................................. 39
     3.6.4 Challenges in RFID Localization ................................................................ 41
     3.6.4 RFID Localization Algorithms .................................................................... 43
     3.6.5 Web Application Backend .......................................................................... 51
  3.7 DATABASE ............................................................................................................ 54
  3.8 WEB APP FRONT-END .......................................................................................... 55
     3.8.1 Main Flow ..................................................................................................... 55
     3.8.2 Data Visualization ...................................................................................... 57
  3.9 DEVELOPMENT TESTING .................................................................................... 59
  3.10 USER PROFILE .................................................................................................. 60
  3.11 TECHNOLOGY STACK ........................................................................................ 60
     3.11.1 Software Technology Stack ...................................................................... 60
     3.11.2 Hardware Technology Stack .................................................................... 62

CHAPTER 4: DESIGN PERFORMANCE ............................................................................ 64
  4.1 PERFORMANCE .................................................................................................... 64
  4.2 LIMITATIONS ....................................................................................................... 66
  4.3 DIFFICULTIES ENCOUNTERED ............................................................................. 68
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1.1:</td>
<td>MTR PASSENGERS DATA ACROSS THE YEARS</td>
<td>10</td>
</tr>
<tr>
<td>FIGURE 1.2:</td>
<td>DELIVERY MAN IN HKU MTR STATION</td>
<td>11</td>
</tr>
<tr>
<td>FIGURE 2.1:</td>
<td>FLOW CHART OF AN IPS</td>
<td>14</td>
</tr>
<tr>
<td>FIGURE 2.2:</td>
<td>WORKINGS OF IPS MODULES</td>
<td>15</td>
</tr>
<tr>
<td>FIGURE 2.3:</td>
<td>MAGNETIC FIELDS IN BUILDINGS</td>
<td>16</td>
</tr>
<tr>
<td>FIGURE 2.4:</td>
<td>CISCO IBEACON TECHNOLOGY</td>
<td>17</td>
</tr>
<tr>
<td>FIGURE 2.6:</td>
<td>WORKINGS OF A GPS SYSTEM</td>
<td>19</td>
</tr>
<tr>
<td>FIGURE 2.7:</td>
<td>WORKINGS OF A RFID READER AND TAG</td>
<td>22</td>
</tr>
<tr>
<td>FIGURE 3.1:</td>
<td>SENSING SURFACE</td>
<td>23</td>
</tr>
<tr>
<td>FIGURE 3.2:</td>
<td>ATID AT288N READER</td>
<td>26</td>
</tr>
<tr>
<td>FIGURE 3.3:</td>
<td>TAG CONFIGURATION FOR SENSING SURFACE</td>
<td>27</td>
</tr>
<tr>
<td>FIGURE 3.4:</td>
<td>FIRST SCREEN OF THE APP</td>
<td>28</td>
</tr>
<tr>
<td>FIGURE 3.5:</td>
<td>BLUETOOTH PERMISSIONS</td>
<td>29</td>
</tr>
<tr>
<td>FIGURE 3.6:</td>
<td>FUNCTION FOR CONNECTING READER</td>
<td>29</td>
</tr>
<tr>
<td>FIGURE 3.7:</td>
<td>SECOND SCREEN OF APP</td>
<td>30</td>
</tr>
<tr>
<td>FIGURE 3.8:</td>
<td>THIRD SCREEN OF APP</td>
<td>31</td>
</tr>
<tr>
<td>FIGURE 3.9:</td>
<td>FOURTH SCREEN OF APP</td>
<td>32</td>
</tr>
<tr>
<td>FIGURE 3.10:</td>
<td>FUNCTION TO SEND TAG DATA TO DB</td>
<td>33</td>
</tr>
<tr>
<td>FIGURE 3.11:</td>
<td>PASSIVE RFID TAG</td>
<td>38</td>
</tr>
<tr>
<td>FIGURE 3.12:</td>
<td>ACTIVE RFID TAG</td>
<td>39</td>
</tr>
<tr>
<td>FIGURE 3.13:</td>
<td>FRIIS TRANSMISSION EQUATION</td>
<td>44</td>
</tr>
<tr>
<td>FIGURE 3.14:</td>
<td>BAYESIAN INFEERENCE EXAMPLE</td>
<td>46</td>
</tr>
<tr>
<td>FIGURE 3.15:</td>
<td>BAYESIAN NETWORK FOR LOCALIZATION OF MOBILE TARGETS</td>
<td>48</td>
</tr>
<tr>
<td>FIGURE 3.16:</td>
<td>MULTILATERATION ALGORITHM</td>
<td>48</td>
</tr>
<tr>
<td>FIGURE 3.17:</td>
<td>LOCATING READER WHEN READER RECEIVES SIGNALS FROM TWO TAGS</td>
<td>50</td>
</tr>
<tr>
<td>FIGURE 3.18:</td>
<td>LOCATING READER WHEN READER RECEIVES SIGNALS FROM THREE OR MORE TAGS</td>
<td>51</td>
</tr>
<tr>
<td>FIGURE 3.19:</td>
<td>PSEUDOCODES OF BACKEND API</td>
<td>53</td>
</tr>
<tr>
<td>FIGURE 3.20:</td>
<td>BASIC DATA SCHEMA OF THE PROJECT</td>
<td>54</td>
</tr>
<tr>
<td>FIGURE 3.21:</td>
<td>SIGN-IN VIEW</td>
<td>56</td>
</tr>
<tr>
<td>FIGURE 3.22:</td>
<td>MAIN PAGE VIEW (NO VIOLATIONS)</td>
<td>56</td>
</tr>
<tr>
<td>FIGURE 3.23:</td>
<td>MAIN PAGE VIEW (WITH VIOLATIONS)</td>
<td>57</td>
</tr>
<tr>
<td>FIGURE 3.24:</td>
<td>SHOP VIEW</td>
<td>57</td>
</tr>
</tbody>
</table>
List of Tables

TABLE 3.1: READER LOCALIZATION VS. TAG LOCALIZATION ................................................................. 37
TABLE 3.2: ACTIVE AND PASSIVE TAGS .......................................................................................... 38
TABLE 3.3: TAG FREQUENCY .......................................................................................................... 41
List of Abbreviations

MTRC – Mass Transit Railway Cooperation
EMSD – Electric and Mechanical Services Department
IPS – Indoor Positioning System
GPS – Global Positioning System
HMI – Human Machine Interaction
SLAM – Simultaneous Localization and Mapping
RFID – Radio Frequency Identification
INS – Inertia Navigations Systems
EPC – Electronic Product Code
RAM – Random Access Memory
ROM – Read Only Memory
CE – Common Emitter
API – Application Programming Interface
FCC -Federal Communications Commission
LF – Low Frequency
HF – High Frequency
UHF – Ultra High Frequency
RSS – Received Signal Strength
OTP – One time Password
Chapter 1: Introduction

This chapter is the introduction to the final report for the project pertaining to in-station delivery service safety monitoring. It provides the context and problem statement, followed by objectives, scope and deliverables as required by MTRCL, and lastly, the outline of the report.

1.1 Background

Mass Transit Railway, more commonly known as the MTR, is a robust public transport network in Hong Kong. It is operated by the MTR Corporation Limited and consists of light and heavy rails running covering 237km of track spanning over 90 stations [1]. On average, it makes nearly five million passenger trips daily, making it one of the busiest modes of public transport in the world. [3] This huge system is managed by the Mass Transit Railway Cooperation (MTRC) which looks over the functionality, maintenance and safety of the entire system. Apart from serving as a platform for the trains, the MTR stations are home to almost 1500 shops which are a key part of the customer experience provided by the MTRC. These shops provide good and services including food, beauty, fashion, shoe repairs and travel services. [1]

![Figure 1.1: MTR Passengers Data across the years](image)
1.2 Motivation

The 1500 stores in the MTR stations require frequent deliveries to their replenish stock. These deliveries are carried out during the working hours of the MTR, i.e., during the same period of time MTR stations receive passenger footfall. Delivery personnel use the same pathways, entries and exits as the passengers. The goods are packed into cartons, which are stacked on top of each other and transported to the stores via a trolley. These bulky cartons may be stacked as high as 2m or more, which effectively blocks the vision of the delivery personnel. Furthermore, navigating trolleys through restricted areas and/or at very high speeds jeopardizes the safety of passengers. The Electrical and Mechanical Services Department (EMSD) gauged this situation as a risk to public safety and called for it to be mitigated.

![Delivery Man in HKU MTR Station](image)

Figure 1.2: Delivery Man in HKU MTR Station

1.3 Objectives

The project aims to mitigate the risk of in-station delivery described above by introducing a sensor-based monitoring system based on RFID technology to detect malpractices of delivery personnel. On detection of any violation, the relevant authorities in the MTR station would be alerted and the
relevant statistical data generated would be tracked for analysis. The MTRCL has laid the
requirements the system must fulfil:

1. Monitor the height of stacked goods does not exceed 1.2 meters
2. Monitor the speed of delivery personnel does not exceed 1.3 meters per second
3. Monitor unauthorized entry into restricted areas
4. Alert mechanism in the event any of the above violation is committed

With such a system in place taking care of the above requirements, the MTRCL hopes the
malpractices will be abated, and the safety of the passengers will not be compromised.

1.4 Scope

The scope of this project is limited to objectives 2,3 and 4, i.e. speed detection, area violation and a
holistic alerting system. After conducting meetings with the MTRCL it was decided that due to
limitations pertaining to data protection, the project will be unable to leverage the CCTV recordings
of the stations. This prevented the use of Machine Learning techniques (ex. object detection, computer
vision) to detect height of stacked goods from live video footage. Moreover, the refusal to install
bulky infrastructure restricted the use of line-of-sight sensor-based systems for height detection as
they are being transported through the station. Thus, with all viable methods to solve the problem
using hardware or software solutions deemed off-limits, the fulfillment of objective 1 was
unachievable.

Apart from the core requirements, the project will also deliver a real-time application to send
statistical data of the violations and violators to the MTRCL for analytical purposes.

Thus, the scope of the project entails the following deliverables:

1. The first and foremost deliverable is the real-time monitoring and alert system. The unique
   restrictions and requirements of the system entail that it can be broadly categorized as an
   implementation of an Indoor Positioning System (IPS). Indoor Positioning System is a term used to
decribe a network of devices used to precisely locate people or objects in an indoor location. [3]

2. Secondly, the project will extend beyond the core requirements and provide a statistical tool in the
   form of a web application to alert the MTRCL authorities if a violation has been committed and
   provide basic data analysis for further assist in mitigation of risk.
Note: The project will aim to produce a Proof of Concept backed with sufficient evidence and findings to ensure that the design and implementation can be efficiently scaled to all MTR stations.

1.5 Outline

The remainder of this progress report is divided into the following sections:

1. **Literature Review**: This section presents an overview of the writings and prior research done in this area with a summary on each writing to point way forward to future research

2. **Design**: This section explains the methodology and implementation of the solution along with in depth analysis of the design choices and their justification.

3. **Design Performance**: This section is responsible to give the analysis of the solution and lay down the testing process, limitations and the difficulties encountered

4. **Recommendations**: This part of the report lays down the recommendations for future researches who might be interested in this topic and provides advices to them.

5. **Future**: The penultimate section of the report lays foundation for the future work to be completed in the project with an emphasis on the things that can be improved.

6. **Conclusion**: The final section of the report lays down a conclusion and a summary of the entire report.
Chapter 2: Literature Review

This chapter gives an overview of the writings and sources on IPS and localisation algorithms. It provides a summary of the sources to point the way towards further research.

2.1 Indoor Position System

Navigation devices assist users in navigating new environments. The recent technical advances have driven users to encapsulate these systems in handheld devices, effectively increasing the prevalence and number of users using these systems. The lack of Global Positioning System (GPS) signals indoor allows indoor navigation more difficult than in outdoor environments which is why radio frequency signals, sensor-based devices and computer vision are the preferred solutions for indoor positioning.

![Figure 2.1: Flow chart of an IPS](image)

A human indoor positioning systems is made using 3 modules – Indoor positioning module, Navigation module and Human Machine interaction module. The indoor positioning module is used to find the user’s coordinates, the navigation module is used to get the route from Point A and Point B and the Human Machine Interaction (HMI) module helps in interact with the above 2 modules and send instruction to the user.[4]
Diving deep into Indoor Positioning Module, several devices and techniques are used for navigation. They can be broadly classified into Magnetic Positioning, SLAM, radio waves, Bluetooth, mobile sensors and RFID technology.

1) **Magnetic Positioning**: Animal navigation was the inspiration for magnetic positioning. Animals like migratory birds use the Earth's magnetic fields to find their way to their destinations. Similarly, smartphones detect and react to magnetic field variations within buildings. Using these variations, smartphones detect the position. These variations are calculated using the magnetic flux density which is defined as the magnitude of magnetic field over a surface. It then uses positional algorithms like Trilateration and Kalman Filter Algorithm to give an estimate of the position. These fingerprints in building need not necessary be of the building but can be crowdsourced also. For example, common people like walker byes, store staff can generate magnetic fingerprints using applications on their smartphones. One such application is Waze developed by Google. One downside of this technique is that these magnetic anomalies must be calculated well in advance. [5,6]

Some applications of Magnetic positioning have been done in [7,8,9]
2) **Simultaneous Localization and Mapping (SLAM):** SLAM is the technique behind robotic cartography and robot mapping. It creates an indoor map using unmanned vehicles or sensors and can be used for both 2D and 3D motion. [5] These systems are mobile that is they are best suitable when the user moves. It cuts GPS altogether and allows the user to reach complicated enclosed spaces opening a whole new environment to be surveyed and mapped. There are various types of SLAM algorithms – Graphs SLAM, EFK Slam, Fast SLAM, Topological SLAM, Visual SLAM, 2D LiDar SLAM, 3D LiDAR SLAM and ORB SLAM. However, since it’s an up and coming technique, its highly complicated and doesn’t have a large community support. [10] Some applications of SLAM are done in [11,12,13]

3) **BLE Beacons:** The latest entry into indoor location tracking are Bluetooth Low Energy beacons. These beacons are able to calculate the distance between themselves and any smartphones/iBeacon that enter its range of detection. They were developed by Apple in 2013 and were originally limited to iOS devices but can now be used by third parties to develop their own SDKs. For example, PayPal is already creating their own cheaper iBeacon-compatible hardware. The major advantage this hardware possess is its small size, ease of adaption and a widespread community. [14] They are primarily used to detect a user’s entry and exit which is why even the MTRCL tested a solution involving iBeacons in 2019 to mitigate the risk due to in-station delivery. With the ability to track locations of a user, the iBeacons technology was thought to be successful, but to a certain degree. It could only
monitor the speed of the delivery personnel (objective 2) and provided no means to fulfil the other objectives.

Another example of iBeacon technology being developed by third parties is the CISCO SDK which is describe in the figure below. They have also been used successfully in [15,16,17]

Figure 2.4: CISCO iBeacon Technology

4) **WiFi:** WiFi can be used in the same manner as beacons to track location with an accuracy of about 5-15 meters. However, though its signal is better and can go further than BLE, it needs an additional power supply, expensive infrastructure and higher setup costs. Furthermore, things such as the human body and walls can interfere with WiFi signals and the opening and shutting of doors may have an effect on WiFi signals.[5]

5) **RFID:** RFID, or radio frequency identification, is a technology that consists of a tag and a reader. The reader reads the data in the tag using a radio-frequency electromagnetic field to identify the source to which the tag is applied. The tag can be active or inactive depending on whether it has a battery. It can also be used in conjunction with other technologies to provide precise indoor position. This project uses RFID and the details of it will be given later in the report. Successful application examples are [18,19]

Apart from the above technologies, sensors and radio waves like ultrasonic have also been used in the past.
2.2 Failure of GPS in an IPS

Nowadays, GPS is the one stop solution for location tracking since it is free and easy to use. Its convenience plays a huge role in its adaptability and widespread usage. GPS works through a chip installed in your mobile device which weighs less than 5 grams [20] and acts as a receiver for the signals sent through a multi-million-dollar network of satellites revolving around the planet.

GPS was launched in the year 1978 but became fully operational in 1993 when a robust satellite system was setup. Each of these satellites consist of atomic clocks and broadcasts signals which consist of its time and position. These signals are caught by the receiver in our phones to give an estimate of our location in the form of latitude, longitude and altitude. GPS positioning is based on the concept of trilateration. Trilateration is way to determine position of an object by calculating the distance to points at already known coordinates. In principle, it requires at-least 3 ranges to 3 known points. GPS however requires 4 ranges called as pseudoranges and 4 known points which are the satellites.

Pseudorange is the range (distance) between two points but it takes into account the clock errors that might occur. The clock errors are generated when the time stamp of signal transmission and time stamp of signal reception is recorded. The formula for pseudorange is as follows: [21]

\[
Pseudorange = (\text{Time}_{\text{signal-sent}} - \text{Time}_{\text{signal-received}}) \times \text{(speed of light)}
\]

The satellites send their atomic time in the signal to the receiver. The receiver then uses the pseudorange calculated with the formula described above to calculate its own three-dimensional position which is the further used for location tracking. [21]
A standard GPS signal recipient searches the input signal in two dimensions: Doppler frequency and code delay. The searching process locally generates a CDMA code of the satellite and multiplies it by a down-converted trail value of Doppler to get the receiving signal. This process is carried out repeatedly on lumps of incoming signals called as integration periods which is generally 1ms, which means that every 1ms the position is recalculated. [22]

In an indoor environment, spatial problems arise due to lack of time in decoding satellite data. Since local time is not accurately measured due to interferences of walls and building noise, there’s problem accurately calculating position in the defined integration time of 1ms. One solution to this can be to increase the integration time but that results in data bit errors. Experiments have been carried out to increase the integration time till up to 20ms but that has resulted in increasing time for acquiring the signals sent via the satellites. This leads to errors in calculating the pseudorange and in turn is unsuccessful in calculating the location accurately. [22]

Hence GPS fails while being used in an Indoor Position System.
2.3 Choosing a good IPS

Since numerous methods and technologies have been used to successfully implemented IPS, an evaluation criterion was determined to help evaluate the efficacy of each technology. The major factors are described below: [23]

1. **Accuracy**: One of the most important efficiency metrics for a navigation device is accuracy. This metric is mostly aligned with the navigation system's indoor positioning module. Localization error is expressed in terms of its accuracy. It is calculated as the average Euclidean interval between the ground truth and approximate position coordinates.

2. **Precision**: It’s is defined as the percentage of accurate locations returned. It is concerned with the accuracy of device results or positioning over time and in different situations. The cumulative distribution function can be used to describe the system's precision.

3. **Scalability**: The system's scalability can be assessed by taking two factors into account: geography and user count. Geography denotes the area of the indoor environment. A rise in the number of users in the same area may cause positioning uncertainty due to interference from communication technology-based systems' signals. The solution selected should be independent of MTR station area, floors and infrastructure considerations.

4. **Adaptability**: The potential of a device to survive unfavourable events, such as subsystem malfunctions and signal failure, is referred to as its adaptability. If one or two infrastructure elements crash or malfunction, the solution should still be able to navigate and monitor the user. It should be effective despite changes implemented to all or some MTR stations.

5. **Affordability**: The cost of the solution can be classified into its three modules – navigation, positioning and its HMI module. The cost of a positional module in particular is the cost of the infrastructure components and their maintenance cost. The cost of the navigation modules is linked to the expenses in map construction methods. The HMI cost does not account for much since it is user driven. Hence, the solution in both deployment and maintenance of solution must cost effective.

Other factors that should also be kept in mind are:

6. **Wearability**: The solution must contain hardware devices that are non-intrusive

7. **Environment**: The solution should be unaffected by MTR station materials and signals.
8. **Latency**: Latency is defined as the delay in sending data from one device to another. The solution should be capable of alerting relevant authorities in real-time which is why a minimal latency solution will be preferred.

9. **Power**: The solution should require a minimal power supply.

### 2.4 Efficacy of RFID

The use of Radio-Frequency Identification (RFID) technology proved to be the most ideal for designing an IPS suited to the MTRCL’s requirements. Before discussing RFID technology, the alternate technologies discussed above and the shortcomings they presented are discussed below [23]:

1. **Wi-Fi**: Its accuracy is about 5-15m, but performance is significantly reduced in dense indoor environments. Moreover, change in network could require recalibration of the system, resulting in high cost of maintenance.

2. **Bluetooth**: Cost ineffective as one master device can connect to only seven slave devices, and any slave device can connect to only one master device. This restricts the ability to track across large areas and allows tracking of only one entity. Further, hopping between devices may take up to 10 seconds, failing to accomplish real-time location awareness.

3. **Infrared Systems**: Cheaper alternative but they are line of sight devices which are sensitive to light and require high costs for installation and maintenance.

4. **Ultrasound**: Cheaper alternative but they are also line of sight devices requiring dense setups, leading to high installation and maintenance costs. Further, they pose hazardous effects to human health.

5. **Inertia Navigation Systems (INS)**: INS refer to sensors of various kinds that track physical parameters, such as those applied in GPS navigation and location systems. Most of the systems which use INS today are prototypes to help in GPS navigation. Drawbacks include susceptibility to environmental factors and loss & inaccuracy of data.

6. **Magnetic Positioning**: Accurate and cheap, but accuracy is greatly reduced due to presence and movement of metallic objects (ex. elevators, trains) in the environment.

7. **SLAM** – Less support to extend to all MTR stations and Hong Kong and precision is less if robotic mapping is inaccurate.
On the other hand, RFID technology is primarily used for tracking and detection, which is well suited to the needs of the project. RFID tags are used in numerous industrial applications and are a topic of extensive research. They have proven to be an accurate and precise tool with low latency. A solution using RFID tags can be easily reproduced across all MTR stations, and changes in infrastructure could require a new tag configuration but would not be a setback to its functioning. RFID tags possess the shape and form of a card, and readers that of a handheld mobile device. Noisy signals pose a threat of interference but can be mitigated by auditing the spectrum of frequencies. RFID tags cost between a few cents to a few tens of dollars depending on volume and sophistication. Passive RFID tags do not need any source of power, the only power supply needed would be for the handheld RFID readers, which can be recharged on a daily basis, like smartphones [23,24]

Figure 2.7 provides a visual description of how RFID works. As seen in the figure, the RFID setup consists to two main aspects, RFID tags and RFID readers. Tags consist of an antenna and an integrated circuit equipped with memory storage chips. Readers continuously send radio waves in all directions, eliminating the need of line-of-sight. Whenever such a radio wave hits a tag, a circuit on the tag is powered, and it emits a response signal. This response signal, which is also a radio wave, is received by the reader, and data is extracted from it. Modern tags offer up to 128 bits of memory, containing a unique tag ID. A reader can concurrently read and write to multiple tags at once which increase the accuracy of the position since multiple readings can be used to calculate the location.[24,25]

![Figure 2.7: Workings of a RFID reader and tag](image)
Chapter 3: Design

This chapter introduces the concepts and ideas behind the solution, and subsequently provides an in-depth exploration of the implementation of the same. This includes an elaboration of all design choices and their justifications.

3.1 Solution

This section provides a general description of the conceptualized idea based upon which the Indoor Positioning and Alert System satisfying MTRCLs constraints and requirements will be developed.

The system to be developed must be capable of tracking moving entities in indoor environments. The team proposes the following Indoor Positioning System that uses RFID technology to achieve this objective.

The indoor environment that needs to be tracked, such as MTR stations, can be decomposed into sensing surfaces. Sensing surfaces are physical surfaces of the environment that needs to be tracked. Any physical surface, such as walls, floors or ceilings, could be developed into a sensing surface. These sensing surfaces can be further decomposed into grid of smaller surfaces, similar to a coordinate system. Each grid can be referred to as a location unit. Each of these individual location units will possess a unique identification. This will allow the system to uniquely identify physical locations, given the identification of one of the location units of the physical environment to be tracked [23]. Figure 3.1 provides a graphical representation of a sensing surface created on a tiled floor.

Figure 3.1: Sensing Surface
The location of any moving entity, such as delivery personnel, can be identified with the knowledge of what location unit is currently occupied by the entity. Thus, given the technology to stream the information regarding what location unit is occupied by the entity in real-time, the live position of the entity in the physical environment can be found. Any movement made by the entity would result in the entity being located on a different location unit, as the sensing surface is completely decomposed into such location units. Thus, there would be a new stream of data, updating the system with the knowledge of the new location unit that the entity is occupying. These updated location units can simply be mapped back to the physical location represent. Thus, real-time position can be determined, and all movements of the entity can be tracked.

To system leverages the use of RFID technology to achieve the functionality described in the setup above. Each location unit can be represented by an RFID tag. Every RFID tag possesses a unique tag identification in the form of an Electronic Product Code (EPC). This identification data can be used to assign unique identification to each location unit. Thus, a grid of RFID tags would form the grid of location units, and, by extension, a sensing surface. The entity to be tracked, which in our case will be delivery personnel transporting goods and products to the stores inside an MTR station, will have a mobile RFID reader device in their person while they carry out the delivery. This RFID reader will be responsible for identifying what RFID tag it is located nearest to at any given moment. This data will be used to identify what physical location the delivery personnel is currently located at, and also track movements for changes in position.

Thus, the solution essentially requires:

1. Assigning IDs to physical locations
2. Keeping track of which ID corresponds to what location
3. Implementing a mechanism to retrieve this identification
4. Implementing software for real-time tracking of entity location
5. Calculating if violations have occurred based on information retrieved
3.2 Implementation

This section provides a brief overview of the implementation of the solution explained in section 3.1, including both software and hardware aspects.

The implementation of the solution was broken down into multiple stages, based on the software or hardware aspect being developed or built. These stages are as follows:

1. Acquiring hardware and setting up sensing surfaces in environments to be tracked.
2. Developing the mobile application that connects to the RFID reader via Bluetooth.
3. Developing the back end and associated database of the web application to compute whether violations have occurred based on tag data received.
4. Developing the front end of the web application to provide the MTRCL authorities with a analytical dashboard and also serve as a visualization tool.

The following sections describe each of these sections in further detail.

3.3 Digital Map

The indoor environments in which the delivery personnel will be tracked should first be decomposed into sensing surfaces. In accordance with this, the first stage of implementation of our solution also involved creating a virtual or digital map of the tracked indoor environment, constructed out of sensing surfaces.

Since sensing surfaces are nothing but a grid of RFID tags, this entailed that the team had to identify a location and create a digital map of the same by placing RFID tags uniformly across the area. An important factor the team had to address was identifying the read range of the tags. Here, read range of an RFID tag is described as the physical area in space where the tag is able to receive signals from RFID readers and emit a response signal. If the RFID reader is located outside of this read range, the RFID tag will not be able to receive the incoming signals being transmitted by the RFID reader in all directions, and this it will not get powered up and transmit a response signal.

The RFID reader that was used throughout the course of the project was the AT288N, offered by the company Atid (All that identification) [26]. The AT288N RFID reader is shown in figure 3.2. The AT288N has an ARM7 Core processor, with Random Access Memory (RAM) of 64KB and Read Only Memory (ROM) of 256KB. The data capture specifications of the AT288N RFID reader state
that the operating range of frequency for Common-Emitter (CE) amplifier is 865 to 868 MHz it also specifies the reading range to lie between 0 to approximately 3 meters.

Figure 3.2: ATID AT288N Reader

Using this description as an initial benchmark for testing, the team conducted experiments to assess the performance of the RFID reader with two different types of tags. This was done to remove any assumptions or biases and establish ground truths regarding the performance of the hardware equipment, so that the implementation of the solution could be carried out accordingly. From the
experiments, the team found that the AT288N RFID reader was able to read RFID tag responses from a distance of nearly 3 meters only in environments with zero signal interference, or noise. This was identified as an ideal scenario and could not be applied to real-life situations. Further experimentation revealed the true reading range of the RFID reader turned to be roughly between 0.6 to 0.8 meters only. The reading range of the reader also depended on the type and quality of tags being used.

The reading range of the RFID tags that the team was using for the project was established to be just under 1 meter. This was important in order to determine the base unit in terms of size for the individual location units that made up a grid, or the sensing surface. Thus, each location unit was approximately a circular area with radius just under one meter. A visual representation of this setup, or the very first iteration of the digital map consisting of sensing surfaces, can be seen in figure 3.3. The initial setup had minimal overlap, i.e., less than half of the reading ranges of separate RFID tags overlapped.

![Figure 3.3: Tag Configuration for Sensing Surface](image)

### 3.4 Mobile Application

The mobile application used to connect the reader and interact with the database is developed on Android using the API documentation and SDK provided by HKRFID Limited. The application has to be installed on the delivery man phone which they’ll use when they are walking in the MTR stations. As they move across the mapped area, the reader received signals from tags and relayed this data to a custom mobile application, which further forwarded it to the database.
The SDK provided to the project had support for JAVA (Android) and the APIs were also written in the same language. Since the APIs were key to the development of application the team decided to build an Android application so that full support of the documentation could be taken. To build an android app, Android studio was chosen as the preferred IDE which is the official environment for Android App Development. It has loads of features that the team used like:

- Fast emulator
- Unified environment for all Android OS
- Hot reloading
- Github integration
- Extensive testing tools [27]

Below are the various screens of the application with an explanation on what purpose they serve and the functions responsible for the logic

*Figure 3.4: First Screen of the App*
The screen above is the welcoming screen in the application which has three options:

1. **Connect a device** – This is used to connect the AT288N reader which is the default reader of this application.

2. **Connect the new Device** – If there is some other reader apart from AT288N, this option is used to select that device.

3. **Make Discoverable** - Changes the device’s BT module to connect to its wait state.

4. To connect a device, the user taps on the connect a device and using Bluetooth the reader is connected to the device. This is done via permissions specified in the AndroidManifest.xml file and the connectReader() function.

```xml
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

*Figure 3.5: Bluetooth Permissions*

```java
// Connect Reader...
private void connectReader() {
    reader.connectMostRecentDevice();
    if (reader.mDeviceAddress != null) {
        showDialog(title: "", "Connecting to device. Please wait...");
    }
}
```

*Figure 3.6: Function for connecting reader*
After successful connection with the RFID reader, the screen above shows all the buttons which are now clickable along with the device version. To see the readings of the reader, user can click on the Inventory button which leads to the following next screen. The other buttons on the screen are used for other functions –

- **Inventory (With Memory) and Stored Inventory** – It has the same functionality as the Inventory option but also shows previously stored readings using the Stored Inventory option.

- **The Lock/Kill button** locks the RFID reader so that it cannot be used for reading. This is done so that we can have a handsfree experience with the reader and all its functionalities can be done via the application.

- **The Option button** opens the menu which has options to increase the read time and idle time of the RFID reader connected.
This is the inventory page which is empty since no readings have been read yet. To instruct a reader to start reading, a user has to simply tap on the Inventory button and the reader will be automatically activated.
When the RFID reader reads the tags, they are displayed on this screen with the IDs being the long string on the left. The number on the right shows the number of times the tags have been read and the yellow number on the bottom left shows the number of unique tags read. This screen interacts with the Database and sends the readings displayed to the DB using the following function. A simple post request of the tags read is done to the DB which then sends data forward to the frontend for the visualization.
3.5 Backend

The back end of the web application is the part of the application that contains all the code that controls the processes and logic that needs to operate between the client making a request and receiving a response. This includes server-side code, routing, and database logic. The functioning of the back end of the web application developed by the team for the project is explained in the following paragraphs.

The back end of the web application implements an Application Programming Interface (API). An Application Programming Interface is software that enables communication between applications. APIs allow an application to retrieve from and send data to other applications and provides abstraction, i.e., the application does not need to have any knowledge of how the software being interacted with is implemented.

For the purposes of the project to create an Indoor Positioning System, the backend implements an API that communicates with the remote database and retrieves the information stored in it. To elaborate, the RFID reader which is attached to the moving entity (delivery personnel) receives
response signals from RFID tags while the entity is moving across the tracked area. The RFID reader sends this data to the mobile application. The mobile application further sends this data to the remote database. The API implemented in the backend retrieves this data that is stored in the database.

The data retrieved is checked to ensure it contains all necessary information, i.e., the data is validated. This is a security measure that ensures the code does not break due to some errors caused because of incorrect data types or missing information. This practice ensures that the logic encoded in the program has the data in the type and form that it expects. In short, the data is checked for accuracy and quality. Once the data is validated for correctness, it is normalized. Data normalization is the process of changing the units or values of the data to a fixed standard that the application has been programmed to work with. This practice also allows the program to reject unnecessary and extra data points that the application does not require. The benefits of data normalization are many-fold. The data received as a response from the API may be of unknown type and form. Through the process of data normalization, the structure of this unknown data can be transformed into a structure that the application is programmed to work with. Elimination of data points that are not required also help increase the speed of computation as it reduces the amount of computation the application needs to do to provide outputs.

The backend of the project’s web application has been programmed to compute the location of the moving entity, such as a delivery personnel, in real-time. Localization using RFID tags cannot be done simply on the basis of the received signal sensing, i.e., it cannot be said with certainty that a moving entity is located near a particular reader or tag (depending on the method of implementation, further discussed in the following section), simply because the latest reading of signal response or transmission was corresponding to that device. This is because RFID tags and readers are susceptible to interference, i.e., noise or undesirable signals. They can also perform poorly and inaccurately due to blocking or absorption of the signal. Thus, sophisticated localization algorithms must be implemented, and outlier readings and noise detected by the system must be handled and eliminated to produce high quality results.

Furthermore, it also contains logic that computes whether this entity is committing any sort of malpractice, as stated in the requirements. Firstly, this includes over-speeding of delivery personnel, wherein the speed is greater than or equal to 1.3 meters per second. Secondly, it includes monitoring entry or traversal of the delivery personnel into unauthorized or restricted areas of the tracked environment. Thus, the back end of the web application is responsible for localization and detection of violations.
In the event that the back end identifies a violation, the information corresponding to this violation is accordingly logged. The back end updates the remote database with the information pertaining to the violation immediately, to ensure the system can operate in real-time and provide live notifications to the MTRCL authorities regarding any violations taking place in the MTR stations. The violations being handled are of two types, and the document containing information related to the violation is unique for both types. The first type of violation, i.e., over-speeding contains the following details: tag ID, previous tag ID, speed, timestamp. The second type, i.e., unauthorized area violation contains only the following details: tag ID and time. The choice of database and related schemas will be discussed in more depth and detail in the upcoming sections.

3.6 Localization

Localization was one of the two primary focus of the project, along with visualization. Localization in engineering in technology refers to the process where a device is used to identify or compute the position of a person or object in a real environment [28]. Indoor Positioning is a sub-topic under the broader topic of localization and refers to the in indoor environments. The project developed builds an IPS system using RFID technology. RFID localization has been researched extensively, with promising results. To produce an optimal Indoor Positioning System, the following factors were considered:

1. Reader and tag localization
2. Active and passive tags
3. Active and passive readers
4. Tag frequency
5. Challenges in RFID localization
6. RFID localization algorithms

Each of these factors considered are explored in detail below, and justifications of choices made are outlined as well.
3.6.1 Reader and Tag Localization

Localization using RFID technologies can be implemented in two methods, namely reader localization or tag localization. As the names suggest, they differ by what component of the RFID setup is being used as the set of reference nodes and what component is attached to the moving entity and traverses across the tracked area.

Reader localization is when the entity to be tracked possesses a reader. Tags are placed in the surrounding indoor environment as reference nodes. On the other hand, tag localization is the opposite, i.e., readers are installed in the environment and a tag is affixed to the tracked entity. In this case, readers act as the reference nodes.

The methods of implementation of localization using RFID technology, i.e., both tag and reader localization provide similarly high accuracy. However, tag localization techniques have a lower threshold for speed [29]. This means that given the same hardware equipment, if the devices were used to create a tag localization setup, their performance when the moving entity traverses the tracked area at speeds greater than a certain threshold would be less accurate as compared to the results when the same equipment is used for reader localization setups. The threshold speed above which the system begins to falter is dependent on the quality of hardware.

In terms of cost, the difference between the pricing of RFID tags and RFID readers is quite significant.

RFID tags come in range of different types. The various types are described below [30]:

1. Passive tags
2. Metal passive tags
3. Active tags

Ordinary passive tags are the cheapest and cost around 10 cents each, i.e., 0.10 USD. They can be used on non-metal surfaces.

Metal passive tags are slightly more sophisticated than basic passive tags and can be used in metal applications as well. Thus, they also cost more than the simpler passive tag counterpart. Metal tags are priced around USD 1 per tag.

Lastly, the active tags are the most complicated devices among the three types of tags. They are powered by their own portable battery and are capable of executing commands such as uploading
data to the cloud [31]. They are the most expensive among the three types of tags and can cost anywhere in the range of USD 15 to 20 per piece.

The project implements reader localization. The reason for this is twofold. These reasons are each explained more elaborately below.

First, an implementation involving tag localization would require the entire MTR station be covered with RFID readers. This would highly cost-ineffective when scaled up. Considering there are 95 MTR stations, with varied sizes and areas, different number of floors and entrances/exits, implementing tag localization would require a large amount of RFID readers [1]. Apart from upfront installation costs, this would also entail incurring costs of software and maintenance.

Furthermore, installing such readers would involve a lot of infrastructure overhead. RFID installation must only be done by individuals well versed in the task, as there are multiple factors to consider for the system to run optimally and as desired. These include and are not limited to factors such as tuning of equipment, identifying optimal orientation of readers, directing the antennas, accordingly, running tests to ensure system functions as expected, configuring and adjusting settings so they work for the specific environment, which in the projects case is unique MTR stations, and networking details. The MTRCL had provided one restriction in the form of avoiding installation of bulky infrastructure, and thus tag localization seemed like an undesirable alternative out of the two.

Table 3.1 summarizes the comparison discussed above in a tabular format

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>Cost</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag localization</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Reader localization</td>
<td>High</td>
<td>Low</td>
<td>Lower than tag localization</td>
</tr>
</tbody>
</table>
3.6.2 Active and Passive Tags

As discussed in the previous section, RFID tags come in three different formats, namely, passive tags, metal passive tags, and active tags. An example of each of two of these types, i.e., active and passive tags, can be seen in figure 3.11 and 3.12. Passive tags are the cheapest of the three, costing around ten cents per piece, while active tags are the most expensive, costing around 15-20 USD per tag [30]. Passive RFID tags have a very small read range of anywhere between 10 centimeters (4 inches) to approximately 1 meter. On the other hand, active tags have a large read range ranging of up to 100 meters. Modern solutions and RFID technology industrialists are also inventing active RFID tags that have a range in the unit of a limited number of kilometers, i.e., a few thousand meters [32]. Active tags are battery powered and are less susceptible to signal degradation caused by environmental changes, noise, and orientation. However, the advantages of RFID tag did not outweigh the significantly higher price the tags cost. Thus, considering the requirements of the project did not require tags that possess a very large read range, passive tags were chosen [33].

Table 3.2 summarizes the comparison discussed above in a tabular format

Table 3.2: Active and Passive Tags

<table>
<thead>
<tr>
<th>Type</th>
<th>Read Range</th>
<th>Cost</th>
<th>Signal Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive</td>
<td>10cm – 1m</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Active</td>
<td>Up to 100m and more</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Figure 3.11: Passive RFID tag
3.6.3 Tag Frequency

Apart from being classified as active, passive and metal passive with regards to how the RFID tags receive their power to function, wherein active tags contain their own portable battery and passive tags are powered by the incoming signals transmitted from the RFID reader in the infrastructure, the RFID tags can also be categorized based on another highly important factor. This is explained in the following paragraphs.

RFID tags can be classified into various types based on the frequency they are designed to operate on. Radio frequency is a segment of the large range of frequencies present in the electromagnetic spectrum [34]. The consideration of what frequency the RFID infrastructure and solution should operate on is highly important and there are multiple government authorities that act as regulatory bodies, identifying and allocating resources on the electromagnetic spectrum dedicated to the working and operation of radio frequency-based technology. One such regulatory body is the Federal Communications Commission (FCC) in the United States of America [34].

RFID infrastructure can be classified into 4 categories-based frequencies they operate on [35]:

1. Low Frequency (LF)
2. High frequency (HF)
3. Ultra-high frequency (UHF)

4. Microwave frequency (microwave)

Each of these types of tags based on frequencies is discussed below.

**Low Frequency** (LF) in the electromagnetic spectrum includes all frequencies ranging between 30 to 300 KHz. However, only 125 KHz and 134 KHz frequencies are allocated to RFID technology [35]. Low Frequency tags are the tags that have been developed and used the longest. This means that they have been adopted in commercial and industrial applications due to the huge potential they offer to solve daily problems at extremely low costs. Low Frequency RFID tags obtain power from the RFID reader, i.e., they fall into the same category as passive tags. When the RFID reader emits an initial signal and these low frequency tags are able to capture these signals, they are powered up and transmit their response signal. This phenomenon is known as near-field inductive coupling in technical terms [35]. Since they do not have any source of power of their own and are only charged to a limited extent by the incoming signals from an RFID reader, they have a very short rage of communication [35]. They also have the lowest rate of data transfer among all types of RFID tags and can hold the least amount of data, i.e., they have the smallest memory storage capabilities [35]. Some industrial and commercial applications of Low Frequency RFID tags include object tracking, animal identification, automotive control, healthcare, and various point-of-sale applications [35].

**High Frequency** (HF) in the electromagnetic spectrum includes all frequencies ranging between 3 to 30 MHz. However, only 13.55 MHz is allocated to RFID technology [35]. High Frequency tags also use the concept of near field coupling to power themselves and function. This is similar to the working of Low Frequency tags. High Frequency tags are more sophisticated than Low Frequency tags, in the sense that they have a slightly faster rate of data transfer [35]. However, High Frequency tags are slower than Ultra-High Frequency tags when compared in terms of rate of transfer of data. High Frequency tags possess a very limited read range of less than 3 feet [35]. Multiple high frequency tags can be read at the same time as these tags may possess some functionality that prevents collisions of two or more tags that happen to emit a response signal that interfere with each other [35]. Common industrial and commercial applications include smart cards and object tracking.

**Ultra-High Frequency** (UHF) in the electromagnetic spectrum includes all frequencies ranging between 300 to 1000 MHz. However, only 433 MHz and the range of frequencies lying between 860
to 890 MHz are allocated to RFID technology [35]. Ultra-High Frequency tags generally are able to read and emit signals up to a few tens of feet. Ultra-High Frequency tags are equipped with some sort of protection against tag collisions.

For the implementation of the project, considerations for what type of RFID tag to use based on the frequency it operated on was as follows. Since a digital map had to be created, with each location unit representing some real physical space, similar to a coordinate system, the team decided to make use of low frequency RFID tags as they would ensure a more holistic coverage of the MTR stations and the location and tracking of the delivery personnel could be done with a higher degree of accuracy. Furthermore, leveraging the use of Low Frequency RFID tags also provided the benefit of being cheaper and making the entire project more cost-effective.

Table 3.3 summarizes the comparison discussed above in a tabular format.

<table>
<thead>
<tr>
<th>Type</th>
<th>Read Range</th>
<th>Memory</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Frequency</td>
<td>Less than 1 foot</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>High Frequency</td>
<td>Less than 3 feet</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Ultra-High Frequency</td>
<td>Less than 30 feet</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

### 3.6.4 Challenges in RFID Localization

RFID technology has been experimented with and research to a great degree. It has many real-life applications, including but not limited to localization of objects and persons. This localization technique that makes use of RFID technology has proven to have great potential and has been used to develop highly robust systems. However, this does not mean the RFID technology does not contain any drawbacks of its own. This section discusses the challenges faced when implementing a localization solution using RFID technology.
If viewed from a very high level, localization using RFID is fairly straightforward. Regardless of the method of implementation, i.e., reader localization or tag localization, the process to identify the current position of an entity being tracked by the system and also listen for any changes or updates in this position, i.e., track the movements of the entity, the system would simply require triangulation of what RFID reader or what RFID tag is currently being detected, for tag localization and reader localization respectively. This entails that the device that picks up or transmits response signals is the device the entity is located closest to. While this approach or ideology is logical, and to a large extent this is how RFID localization systems are developed, there is need to handle the system and control the inaccurate reading or false signals that the system may pick up on.

RFID devices have drawbacks pertaining to signal reception, strength, and noise. An RFID reader could potentially receive a response signal from a tag that is not part of the positioning and alert system. This means that the RFID reader would be transmitting information across the mobile application and the application would insert this data to the database. On receiving this data point from the API, the back end would not be able to recognize the device (RFID tag) that generated this reading. If not handled properly, this could cause issues in performance of the real-time location tracking system.

Another potential scenario is when the RFID reader picks up on signals from an RFID tag that does belong to the indoor positioning system, but this reading was a chance reading, and the RFID reader was not actually located near that tag. Radio frequency signals are susceptible to noise and factors in the environment that could cause the signal to be unnaturally amplified and travel across distances larger than it normally would. This would provide absurd results as the localization system would show the live position of the moving entity to be highly volatile and random if every tag read was simply taken at face value and registered as the current position of the tracked object or person.

Lastly, radio frequency signal reads are highly accurate in unobstructed space. However, in real-world applications, this accuracy degrades. This degradation could be due to reduction in signal strength because of absorption or blocking [33]. It could also be caused by interference with other signals (such as collisions with other reader or tag response signals) or multiple reflections and refractions during propagation.

Thus, there is a need for implementing localization algorithms for outlier removal and noise detection and removal to ensure robust performance of the system that provides accurate and precise results.
3.6.4 RFID Localization Algorithms

Having established the need for RFID localization algorithms above, this section introduces and explores various algorithms being used in RFID localization systems, and briefly describes the algorithm implemented in this project.

According to the simple RFID localization system model described in the previous section, wherein the mobbing entity (person or object) to be tracked, is simply determined to be located near the position where the latest signal receipt or transmission is detected of an RFID reader (in the case of tag localization) or an RFID tag (in the case of reader localization), the only factor to keep in mind while implementing the system would be the propagation of radio frequency signals through space. The propagation of radio frequency signals can be mathematically described using the Friis equation. The Friis equation is shown below [33, 36]:

\[
P_r = P_t \frac{G_t G_r \lambda^2}{16L\pi^2 d^2}
\]

The Friis equation is also known as the Friis transmission formula and was presented by radio engineer Harald T. Friis in 1946. The Friis transmission formula is used to calculate the power of the signal received by the receiving antenna, for example, an RFID tag, when a signal has been emitted or sent by a sending antenna, for example, an RFID reader. The formula takes into account the distance between the sending antenna and the receiving antenna and also the characteristics of the signal such as what frequency the signal has [37]. In the Friis transmission equation displayed above, \( P_r \) is the power received by receiver antenna, \( P_t \) is the power input to transmitter antenna, \( G_t \) is transmitter antenna gain, \( G_r \) is receiver antenna gain, \( L \) is system loss factor, \( \lambda \) is wavelength, and \( d \) is the distance between transmitter antenna and receiver antenna. The Friis transmission equation shows that rate of decrease of signal strength is inversely proportional to distance travelled [33].

Figure 3.13 shows a graphical representation of the computation that Friis transmission equation carries out.
However, this is ineffective in real-world applications, as discussed in the previous section. Thus, it is important to use localization algorithms for outlier removal and noise detection. Localization algorithms can belong to one of two groups. First, algorithms that calibrate signals in the environment and then estimate position, such as Multilateration and Bayesian Inference. Second, algorithms that compute position through Received Signal Strength (RSS), such as Nearest Neighbor, Proximity, and Kernel based algorithms [33]. The RFID reader acquired for this project did not have the capabilities to measure the strength or power of the incoming signals it receives. Thus, the project implemented localization algorithms for the Indoor Positioning Systems that were of the first type, i.e., algorithms that adopt a two-step approach of calibrating then estimating. The Bayesian Inference and Multilateration algorithms are described below. The Received Signal Strength algorithms are described in the future work section.

**Bayesian Inference Algorithm**

To understand Bayesian Inference, it’s important to know the theorem on which it is based on which is the Bayes’ theorem. In simple language Bayes’ theorem uses prior knowledge of an event to predict the probability of a related event. For example, if we want to know the probability of picking an Ace from a deck of 52 cards knowing that the card picked is red, we can use Bayes’ theorem to calculate the probability. The Bayes’ theorem is as follows:
Here $P(A|B)$ is the probability of $A$ given we have prior knowledge of $B$. $P(B|A)$ is the opposite and $P(A)$ and $P(B)$ is the probability of $A$ and $B$ happening independently.

In our example above is $A =$ picking an Ace and $B =$ red card then we have to calculate $P(A=Ace|Red)$. The other values will be known –

$P (B = Red|A) = 2/4 = ½$

$P (A = Ace) = 4/52 = 1/13$

$P (B = red) = 26/52 = ½.$

Substituting the above values in the equation we get $P(A=Ace|Red) = 1/13$ which is the expected answer.

Now Bayesian inference as a technique is used to get the inference which is the process through which you infer certain properties of a population. It uses Bayes’ theorem to get this inference by using a prior probability distribution and modelling it against an experimental probability distribution to get a third distribution which is used to get the required statistics from. [38]

For example, in the figure below the pink distribution called the posterior distribution is generated using the blue (prior know distribution) and yellow (experimental distribution) distributions.
Bayesian Inference provides direction on how one’s beliefs, or in technical terms, the probability of an event occurring, should be updated when given new or updated data or information [39]. Figure 3.14 shows the fundamental principle behind identifying position of a stationary object using an RFID localization system.

In the figure above, \((x, y)\) is assumed to be the coordinates of the node to be found or located. The other nodes, i.e., nodes represented by \(s_1, s_2, \ldots, s_n\), are the signal strengths received or transmitted by
the node represented by \((x, y)\). Here, an assumption is made that when provided with node \((x, y)\), the received or transmitted signal strengths are independent of one another. Thus, this system satisfies Markov condition [33]. A Markov chain is a mathematical system. This mathematical system describes a series of events that can occur. The change of the system from one state to another state is occurs based on probabilistic rules. The future state of the system depends only on the current state [40].

Thus, following equation can compute the position of the target node:

\[
P((x, y)|s_1, s_2, \ldots, s_n) = \alpha P(s_n|(x, y)) \\
\times P((x, y)|s_1, s_2, \ldots, s_{n-1})
\]  

(3)

Bayesian Inference can also update the probabilistic modal, or belief or what the future states are going to be based on new and updated data that it could potentially keep receiving. This ties in nicely with the application of the project, wherein the state of the moving entity had to be tracked and beliefs regarding where the entity is located had to be updated in real-time.

The following equation represents how Bayesian Inference can compute the location of a moving entity under the Markov assumption [33]:

\[
P(l_t|s_{1:t}, u_{0:t-1}) = \alpha P(s_t|l_t) \int_{l_{t-1}} P(l_t|l_{t-1}, u_{t-1}) \\
\times P(l_{t-1}|s_{1:t-1}, u_{0:t-2}) dl_{t-1}
\]  

(4)

A visual representation of this mathematical equation to track a moving entity using Bayesian Inference can be seen in figure 3.15.
Multilateration

As shown in figure 3.16, Multilateration computes the position of an object by triangulating its position based on reference nodes and knowledge of their coordinates.

![Multilateration Algorithm](image)

Figure 3.16: Multilateration Algorithm

Given a set of reference nodes whose coordinates are known by the system, such as RFID tags strategically located across an indoor environment, in the case of reader localization, the distance between the target node (whose coordinate locations are also known), i.e., the moving entity to be tracked and the reference nodes can be computed using the distance formula given below:

\[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]  

(5)
The equations representing the distance of the target node to all the reference nodes will provide a system of equations which will look something like:

\[
\begin{align*}
    r_1^2 &= (x - x_1)^2 + (y - y_1)^2 \\
    r_2^2 &= (x - x_2)^2 + (y - y_2)^2 \\
    \cdots \\
    r_n^2 &= (x - x_n)^2 + (y - y_n)^2
\end{align*}
\]  

(6)

This system of equations is currently non-linear because we have the $2^{nd}$ power of the ordinates and absiccas of the various reference and target nodes. The system of equations can be linearized by subtracting each alternate non-linear equation from the very initial one. On simplification, we can denote:

\[
b_{i1} = \frac{1}{2}(x_1^2 - x_i^2 + y_1^2 - y_i^2 + r_i^2 - r_1^2)
\]  

(7)

After this step, the linearized system will look like:

\[
\begin{align*}
    (x_1 - x_2)x + (y_1 - y_2)y &= b_{21} \\
    (x_1 - x_3)x + (y_1 - y_3)y &= b_{31} \\
    \cdots \\
    (x_1 - x_n)x + (y_1 - y_n)y &= b_{n1}
\end{align*}
\]  

(8)

This can be written in Matrix form as follows:

\[
A = \begin{pmatrix}
    x_1 - x_2 & y_1 - y_2 \\
    x_1 - x_3 & y_1 - y_3 \\
    \cdots & \cdots \\
    x_1 - x_n & y_1 - y_n
\end{pmatrix},
X = \begin{pmatrix}
    x \\
    y
\end{pmatrix},
\begin{pmatrix}
    b_{21} \\
    b_{31} \\
    \cdots \\
    b_{n1}
\end{pmatrix}
\]  

(9)
Since the team did not have an RFID reader capable of measuring signal strength, and the Bayesian Inference approach, although a two-step approach, still leverages the use of signal strength or power, the team chose to focus on the Multilateration algorithm for the project. Having provided the mathematical explanation above, the algorithm is explored in a more graphical format in the following paragraphs for a deeper understanding.

The algorithm locates the reader based on the number of unique tags the reader is able to receive response signals from. This is elaborated as follows [41]:

1. **Reader receives signals from only one tag**: If only one tag is read, there is insufficient information to carry out any mathematical computations. The location is considered to be within the read range of the tag read.

2. **Reader receives signals from two tags**: Passive tags generally use a dipole antenna and have a doughnut-shaped radiation pattern [42, 43]. For a 2D plane, this can be approximated to a circle. Thus, the patterns from two tags can be described by a system of non-linear equations representing circles. As shown in figure 3.17, solving this system will give the 2 points of intersection of the tag read range circles. The mid-point of the line joining these 2 points is considered to be the location of the reader. The error is distance between midpoint and the points of intersection.

   ![Figure 3.17: Locating reader when reader receives signals from two tags](image)

3. **Reader receives signals from three or more tags**: Expanding the logic from the case where signals from two tags are received, the distance between the reader coordinates and the reference tags form a system of equations (shown above) and can be solved to obtain the readers location. This is shown in figure 3.18:
An important point to be noted is that although Bayesian Inference adopts the approach wherein it goes through two steps, calibration of signals in environment and estimation of location, it does so with the help of signal strength. As mentioned earlier, the RFID reader used to test and implement this project did not have the capability to measure signal strength. Furthermore, since the team could not acquire any formal blueprints for any MTR station due to data protection policies, it was unfeasible to create a virtual map with accurate coordinates. Thus, the team has focused on implementing the localization using a frequency-based alternative to the Multilateration algorithm.

### 3.6.5 Web Application Backend

This section elaborates on the code specific implementation of the backend API that retrieves the tag data from the remote database and the localization algorithm logic it executes to detect and remove outliers and noise.

The backend of the web application consists of an API to fetch tag reads and compute violations. As a pre-requisite, the system requires two collections of data. First, the digital blueprint of the area to be tracked, or the mapping of tag IDs to physical locations. Second, the list of tags that form the digital blueprint, known as validTags. This data is stored in a remote No-SQL Cloud Firestore database.

Every time a new tag, say T, is read by the reader, the tag information is sent to the mobile application. The mobile application inserts this data into the database by hitting an endpoint on the API. On receipt of new tag information, the following logic is run:
First, the software adds a time stamp to the tagID read and then proceeds to check whether a tag containing the same tag ID information as tag T exists in the validTags list. If there is no match, the tag is deemed invalid, i.e., not part of the digital mapping, and no computation is done.

Second, a frequency map of all tag reads registered in the last 1 second is created, i.e., an object that contains information of tag IDs read in the last second along with the number of times they were read. Tag ID of tag T is compared with tag ID of the maximum frequency tag. If it does not match, a document containing tag T’s read time as average time is created and inserted into an averageTime collection. If it does match, a document with tag ID same as tag T’s ID will exist. This document is retrieved, and the average time is recomputed and inserted, using the formula shown below. This step identifies current location of the reader. The last updated tag entry in averageTime is the tag the reader is currently closest to (reader location).

\[
\text{avgTime} = \frac{\text{prevAvgTime} \times \text{count} + T.\text{time}}{\text{count} + 1}
\]  

(12)

Then, the isAuth information in tag T is checked to determine whether the reader is in restricted or authorized area. If the reader is in a restricted area, a violation document is created and inserted into the violations collection. Thus, entry into restricted areas is monitored.

Lastly, on every update of the averageTime collection, speed violations are checked. This is done by retrieving the distance between tag T and the tag inserted before tag T in the averageTime collection and computing the differences in their average times. The formula shown below is used to calculate speed and if speed exceeds the threshold, a violation is logged in the violations collection. Thus, speed of delivery personnel in monitored.

\[
\text{speed} = \frac{\text{distance}}{\text{time}}
\]  

(13)

Thus, the moving entity is tracked, and violations are detected using RFID localization with outlier detection and noise removal.
time = currentTime
tagID = req.body // this is the tag that the reader has just registered

averageTime = DB.collection.get()
violations = DB.collection.get()

if tagID not in validTags:
    reject
else:
    tagsRead[] = fetch tags registered in last 1 second

    // create frequency map
    for each tag in tagRead:
        if tagID exists in frequencyMap:
            frequencyMap[tagID] += 1
        else:
            frequencyMap[tagID] = 1

    maxFrequencyTag = frequencyMap.max()

    if tagID = maxFrequencyTag:
        if tagID in averageTime:
            averageTime.update(tagID, time) // update average time of tagID
        else:
            averageTime.add(tagID, time) // add to avgTime collection

        if tagID.isAuth == False:
            violations.add(tagID, time, isAreaViolation: true)

Figure 3.19: Pseudocodes of backend API
3.7 Database

The project utilizes a No-SQL database for scalability and flexibility. The database is divided into a few data collections for storing data. The database is real-time in nature and data sync is almost instantaneous. These requirements for the database were essential for this project so the decision to choose Firebase Cloud Firestore was made as it fulfills all the criteria. More details of why this choice was made is discussed in the Software technology stack section. This section delves into in depth explanation of each of the data collections seen in Figure 8 and reviews their purposes.

![Figure 3.20: Basic data schema of the project](image)

a) **tags**: Starting with the most basic collection, it simply stores the tagID of every tag that is detected by the mobile app.

b) **validTags**: This collection simply stores all RFID tags that belong to our system and are therefore valid. This is done because RFID is a very common technology nowadays and it is easy to find tags located everywhere which can be detected by our readers. This could cause a problem since these would not be a part of the system and might make it malfunction. Therefore, to keep track of what is and what isn’t part of the system, **validTags** is used. The **isAuth** data field is used to see whether this tag is placed in an authorized area (for deliveries) or not.

c) **averageTime**: This is the most important collection for this project because this is where all the heavy lifting happens. As mentioned in the earlier sections, localization is done by referencing the number of detections of a certain tag in the past unit of time. Simply put, the more times a tag is detected at a point, the more likely it is that that point is closer to the detected tag. **tagID**, pretty intuitively gives the id of the tag which this document refers to,
count is the number of tags which have been used to calculate this average time. And of course, the avgTime field is used to store the average time calculated (calculation mentioned in previous section).

d) violations: As the name suggests, this collection stores data on violations which occur during a delivery session. The time field simply refers to the time at which the violation occurs. There are primarily two kinds of violations which can occur during any session, namely, ‘Area violations’ and ‘Speed ‘violations’. The collection has fields to specify which kind of violation has occurred and supplementary fields to give more information on the violation. If the isSpeedViolation field is true, the tagID, prevTag, and speed fields are populated. The prevTag refers to the previous tag in between which the over speeding was detected (violation detection explained above) and the speed is the speed at which the violation was detected. If the isAreaViolation is true, then no additional information is needed.

3.8 Web App Front-end

The front-end of the Admin we app is built using the React library on top of JavaScript. The technology stack section investigates this in more detail. This section will be further divided into the main app flow and the map visualization.

3.8.1 Main Flow

The main app flow opens with a login screen for authentication. This is just the basic authentication flow used to login a user as seen in Figure 3.21.
After login, we get to the main page, which lists down all the lines of the MTR system as a list. When a line is clicked, a dropdown collapses to show the stations within the line with an option to look at the map of said station or its shops. On the right side, any information on the current day’s violations can be viewed.
If the view shops button is clicked, the shop view opened where a list of shops for the selected station can be viewed as shown in the following Figure. 

**Figure 3.24: Shop View**

### 3.8.2 Data Visualization

With so much data being collected every second, there must be a way to paint a picture of that data so it can be interpreted. Visualization simply gives an interpretation of how to look at the data collected through maps and graphs and charts.
Why is visualization needed? It’s primarily because of how the human brain works. It’s always easier to look at large amounts of data in form of visuals than to sit through pages of spreadsheets. This is because visual mediums are more natural to the human brain and therefore easier to understand and analyze.

The web app uses data visualization in a couple of places, these are:

- **Map**

  To visualize where the delivery personnel is at any given point in time, the app renders a small map which gives a 2D layout of the premises in which the tags have been placed. This was developed using Paper.js. Paper.js is an open-source vector graphics scripting framework that runs on top of the HTML5 Canvas [44]. The reason behind working with this abstraction instead of vanilla JS was that the canvas API is stateless, i.e., it does not maintain references to any paths drawn or shapes created etc. With Paper.JS, every path or curve drawn can be referenced and manipulated later. The map rendered is of the trial location where tested and the tag placements are specific to the location. What this means is for every new location, a new map would have to be created for visualization.
- **Shop Statistics**

Statistics are an extremely useful tool when it comes to data analysis and there is perhaps no better way to represent these stats in form of visuals like charts and graphs. To do this, the team made use of the Recharts library for ReactJS. It is a composable charting library built on React components which is composable, powerful and very reliable (built using SVG elements) [45]

![Shop Statistics visualization](image)

**Figure 3.26: Shop Statistics visualization**

### 3.9 Development Testing

Over the duration of the project, the team has made an effort to follow test driven development wherever possible. All software which has been written has been tested in some form or the other.

Testing libraries used, range from

- React testing library, Cypress and Jest for Web App front-end
- Junit and Mockito for Android App
- Jest and Firebase-testing for back-end

Unit testing was carried out for all software written. Integration tests were written for the mobile app to test how it worked with the backend service. For the web app end-to-end testing was done with
Cypress which is a framework for UI testing in the browser itself (carried out using a variation of a crawler).

### 3.10 User Profile

The system will be used by two users:

1. The delivery men who will use the mobile application so that they can be monitored, and their violations can be recorded. The delivery men will also carry the handheld device responsible for reading the tags.

2. The MTRCL department which will be given full control of the web application along with analytics and statistics for their research purposes.

### 3.11 Technology Stack

This project required both hardware and software support for which careful planning was done to choose the appropriate stack. The team took into account personal preferences and then weighed it with the most suitable technology that could be used to come up with the following software and hardware stack.

#### 3.11.1 Software Technology Stack

Our software application which included the mobile application and the real time alert system web app needed a full stack system for seamless interaction. The following shows the technology used for the mobile application, frontend and backend and DB along with justification on why we used it.

1. **Mobile Application** – The team had various options to choose on how to make the mobile application which included using React Native to make a cross platform application, Swift to make an iOS application and Android (JAVA) to make an Android application. The team finally decided to go ahead with an Android application. This is because the SDK provided to the project had support for JAVA (Android) and the APIs were also written in the same language. Since the APIs were key to the development of application the team decided to build an Android application so that full support of the documentation could be taken.
2. **Backend** – For this project, the backend API services is written in a combination of the Express framework on top of Node.JS and Firebase Cloud Functions. The decision to go with Firebase Cloud Functions was made because the project was already intended to use Cloud Firestore as the database solution and using Cloud Functions would simply make the integration seamless. Another reason for this choice was that Firebase not only provides you the ability write backend code along with some features, but it also let’s allows you host said service on its servers. This way the project was able to cut some cost on hosting. The final reason for the choice was the existence of a local testing suite i.e., Firebase Emulators. This suite essentially has all of Firebase’s most popular services emulated locally for testing purposes so that no additional costs are incurred during testing. The choice to go with JavaScript was made due to its relatively easy learning curve and the team’s experience with the same. Writing the code within the ExpressJS framework also provided a familiar environment and cleaner code.

3. **Database** – The real-time nature of this project required an extremely low-latency real-time database solution. Among hosted databases online, Cloud Firestore was the clear favorite, having support and adequate functionality for the project’s use-case. Alternatives like Firebase Real-time database and RethinkDB were a little obsolete or did not have adequate support for development. Another reason to choose Firestore was the support for live data listeners. What these are is essentially functions which get triggered every time a certain condition within the database is met for e.g., Every time an RFID tag is detected, an update to the speed calculated for the delivery person is triggered. Additionally, since Firestore is also a part of Firebase’s local Emulator suite, it gets extremely easy to run and test with locally.

4. **Admin Web App** – The web app makes use of the React library on JavaScript to make render the UI. This was a relatively simple decision since React is currently one of the most popular UI libraries out there and the team already had extensive experience with the same. For styling and design, we followed Material design guidelines and leveraged it’s React library of the same name. We also used libraries like Paper. js, Recharts for designing the indoor map visualizations using JavaScript’s canvas API and data visualizations.
3.11.2 Hardware Technology Stack

The RFID reader that was used throughout the course of the project was the AT288N. It was chosen after negotiations with HKRFID limited and was the best reader that would suit our needs along with the budget cap imposed by the University. Its small and easy to handle with a battery life of 1 day and an ARM7 Core processor, with Random Access Memory (RAM) of 64KB and Read Only Memory (ROM) of 256KB. The data capture specifications of the AT288N RFID reader state that the operating range of frequency for Common-Emitter (CE) amplifier is 865 to 868 MHz it also specifies the reading range to lie between 0 to approximately 3 meters [26]
Figure 3.28: Hardware Stack

Sends TagID to android application
Chapter 4: Design Performance

This chapter talks about the performance of the solution by talking about the experimental results, testing mechanism, limitations and the difficulties encountered. It gives a holistic report card of the implementation discussed in the previous chapter.

4.1 Performance

To evaluate the performance, the system was iteratively tested in various environments. The purpose of frequent testing was to gain awareness of shortcomings early and act upon them. Such a testing procedure helped the team get an idea about the feasibility of the project.

The initial testing of the reader and the tags were carried out in a stairwell which had no interference of other radio signals and neither did it have any obstacles that might stop the radio waves from being read. The only interference were electric wires, but they were out of the team’s control. The purpose of this test was to get a general idea about the signal strength of the tags and the range of the reader. Since the accuracy of a hardware cannot be always trusted, this test was important to understand the limits of the reader and the tag.

The team then developed an MVP model in March and tested it in more complicated environments by increasing the level of interference. First the MVP was tested at our homes with minimum number of people and interferences the team had no prior idea about. This test showed some shortcomings in our application which included latency issues, logical errors and library inaccuracy on the frontend but provided a good benchmark that needed to be reached. The team then worked on solving this error in theory before the next testing phase began.

The team even ensured checked for tag and reader collisions by creating a simulation using python script and sending request via Postman to bombard the database with huge number of requests to see if our application was resistant to tag and reader collision. Postman is a collaborative API testing tool to make HTTP, REST and SOAP request. It has support to investigate the responses and is easily extendible. [46] With the result, the team was reassured that the system was immune to such collisions and it was ready to be tested out in much more complicated environments.
After the application was built on theory and was ready to be intensively tested, the team took it to environments that looked as similar to the MTR stations as possible. Resemblance was assessed based on volume of moving objects and individuals, and also presence of interference. The project was tested in both outdoor and indoor environment. The tests results were positive – the system was accurately and precisely in detected speed and area violations and provided appropriate alerts for the same. In comparison to existing Indoor Positioning Systems, the system developed was significantly cheaper and did not required complex infrastructure installations. Multilateration requires less computational effort and ensured low latency. Thus, the requirements of the MTRCL were adequately met [33]. The system has numerous other real-life applications, such as elderly person tracking, patient tracking in hospitals, smart homes/malls, museums, walking route analysis in offices and shopping complexes, etc.
4.2 Limitations

Despite the highly positive experimental results, certain limitations have been recognized and are discussed below that can have an impact on the overall performance of the system.

1) **Setup of virtual map** - The first drawback is the setup of the virtual map. The virtual map is the visualization map on the frontend real-time alert web app that enables the admin to see where the delivery man is at in the MTR station. Setting up this map requires physically mapping indoor environments using a coordinate system and feeding the database with the mapping of tag IDs and location units they represent. This will be tedious as there are 95 unique MTR stations to map [1].

2) **Reader-Reader collision** - The second drawback is the system has not been tested for reader-reader collision. In an environment with multiple readers and tags there exists 3 collisions
namely (i) tag-tag collision, (ii) reader-tag collision and (iii) reader-reader collision. Reader-reader collision refers to the situation when signals from multiple readers collide at a tag [47]. This entails that there is currently no guarantee of the accuracy of the system when there is more than one reader, i.e., when more than one entity needs to be tracked. Right now, the project assumes that only 1 delivery will happen at 1 time which means that not more than 1 reader will be present in the station. However, this can be fixed by sending device specific information like reader’s ID along with the tagID to tell the application which reader is sending that specific tag information.

![Reader-Reader Collision](image.png)

**Figure 4.4: Reader-Reader Collision**

3) **Tag Deployment** - The third drawback is that tag deployment strategies have not been explored. The configuration of the tags can have major implications towards the performance of the system. If tag density is high, signal degradation and absorption can be aggravated [47]. The orientation of a tag affects its ability to catch and read signals from the reader, and also transmit strong signals. Ideally, the tags should be perpendicular to the incoming signals from the reader. Optimal tag configurations (minimum number of tags and maximum signal strength) can be identified using greedy algorithms [48].

4) **Privacy of RFID data** – To read data from an RFID signal, a reader is the only device that is required. Hence anyone with an RFID reader can technically catch the readings and see the data. This will cause privacy concerns related to the sensitive data which is why encrypting the data from the tags is of prime importance.
4.3 Difficulties Encountered

Through the course of this project from September 2020 – April 2021, the team was able to successfully finish the project that met all requirements in the scope of the project. However, there were certain limiting factors that harmed improvements and decelerated the speed of work. The inception phase was mostly devoid of any such detriments, but the following were the primary issues faced in the Construction and Testing phase:

- Monetary constraints made it difficult to acquire the RFID readers and tags. This is because most distributors do not sell individual units, and instead provide their clients with the large-scale industrial-level hardware and software, costing over HKD 200,000. The budget of our project was 1000 HKD per person which made it 3000 HKD for the total project. An average industry level RFID reader is priced at around 9K HKD which made it extremely difficult for the team to acquire one. However, after weeks of negotiations with HKRFID Limited, the team was able to get the AT288N reader at a nominal cost and within the budget constraints. The team was also able to acquire 20 passive RFID tags for testing purposes.

- The SDK we received was developed using very early versions of tooling libraries. The team had minimal to no experience in dealing with such software and there was a period of learning before implementation could be carried out for progress to be made. This meant that significant hours of the project were carried out in workshops to understand the API documentation in the SDK so that a suitable application could be built.
Chapter 5: Recommendations

This project has a wide array of features involved which can be amplified and added to. A few suggestions and ideas of how to enhance the project in the future are mentioned in the following section.

5.1 Mapping the MTR stations

To scale up this project in the future, the MTR would have to survey each station and create a layout of the stations in which this system is to be installed. The software will need a 2D representation of station (multiple layouts, for multiple floors) which will be used for visualization. The system will also need a mapping of tags and their positions within the station vicinity. This will be input for the database and will let the app know which RFID tag is placed where in the station.

5.2 Adding Shop/Station specification to the app

As the system this project provides is a proof of concept, it deals with only a certain indoor map. As it is scaled, it can be expected that multiple MTR stations, each having several stores within will be using the app. Therefore, it is necessary to provide a framework to add these specifications to the mobile app at a later stage.

It is proposed the mobile app have a Station and Shop selector before a delivery starts within the premises. The station selected, will inform the app about which indoor map layout to use and the shop detail will inform it about the endpoint of the delivery. The selection of the shop on app start might even help with authentication, which is explained in further detail in the following subsection. As for the real-time app, this will keep a track of which shop within which station is carrying out a stock delivery at the specific moment.

What this means, in terms of the design, is a new set of tags at the delivery checkpoints which will help signal the end of delivery. As for the API, when the shop and station is selected, the API call to the backend will also carry a field indicating which tag will mark the end of this delivery. When this tag is detected by the reader, the web application can indicate that the delivery has been completed and the delivery personnel should now be on their way back.
5.3 Authentication

While the live webapp has authentication set up for organization personnel, the API service which communicates with both the webapp as well as the mobile app does not provide any kind of authentication or security features yet. As a result, anyone with the knowledge of the endpoints of the API can, in theory, access and manipulate the data. This would prove to be a major security breach and therefore, some form of authentication is advised as the system is further developed.

Since the Web App uses Firebase to authenticate users in, the system could potentially use the tokens generated to authenticate the API as well. As for the mobile app, since it is not user dependent, the traditional username/password or biometric authentication cannot be installed.

As an alternative, a user less form of authentication would be preferred, one which would perhaps be dependent on the actual device on which the App is running. The way to identify a specific device is by its deviceID – which the app generates on the first launch and is unique per device.

For this, ‘OpenId Connect’, a protocol to authenticate clients in various ways, based on OAuth 2.0 can be used. Via a client, in our case the mobile app, the user authenticates itself to the server, called the ‘Authorization Server’. The server validates the request, and upon success, issues a token that contains the user identity. The user can then use this token to consume other authenticated services. For our use case, “Resource Owner Password Credentials Grant” [49] is appropriate. Although it uses the traditional username/password model, we can use the deviceID as the username and a generated OTP as the password. The choice for using OTP is made since it is not vulnerable to replay attacks. [50]

It is proposed that a time-based OTP (or TOTP which uses the current time and a shared secret to generate an OTP that is valid only for a very short time range), be generated on the shop’s end every time a delivery is to be made and then relayed to the delivery personnel who can use it to authenticate the app. This OTP can also function as a code to procure the device and reader which will be further explain in the upcoming subsections.

Since the devices will belong to the organization, only the listed devices will be allowed to access the API and this list can be updated as and when needed. This will completely avoid the risk of unauthenticated requests or third-party attacks.
5.4 Concurrency: Multiple deliveries at the same time

Currently, the POC system recognizes only a single delivery happening at once and is not in any way able to handle concurrency. For it to be implemented in a real-world scenario, it needs to be made robust in terms of handling multiple data reads and writes as well as visualizations. To do this it is suggested to make use of a thread-safe data layer to access the database. This would help in race conditions and any data inconsistencies which is extremely important given the real-time nature of the system.

The API also needs to be enhanced to handle multiple requests concurrently so that no server errors occur during run-time. Again, this is of great importance since this is a critical real-time application which if employed, would rely heavily on the reliability of this system and its architecture.

5.5 Hardware enhancements

As a POC, the current system employs hardware with capabilities just enough to prove our concept. As the system is scaled, all hardware components will need to be enhanced as well. The following are some suggestions on how each component may be enhanced to increase the reliability and robustness of the system.

1. **RFID Tags**: As mentioned in section 3.6.3, UHF RFID tags with increasingly high ranges might be used for detection in the future. These tags must be industrial grade to avoid any data loss caused by surface material it is placed on, network interferences etc. Since discretion is also of key importance, these tags will be thin and can be installed within the station premises without much hassle.

2. **RFID Readers**: Industrial grade RFID readers will also be needed to make the system more dependable and scalable. The project presently makes use of the AT288N RFID Reader, which is quite primitive in its features compared to what is available in the market now. Firstly, this reader only operates as a Boolean set-up, i.e., only gives feedback when a tag is detected, and gives no further information. With a reader which gives the Received Signal Strength (RSS) (as discussed in the following subsection), the system can make use of higher accuracy algorithms made with a combination of weighted averages, RSS rank [51], and Generalized Regression Neural Networks (GRNN) [52] to have a better estimate of the localization of the reader. It is suggested that the new reader has a better range to optimize the tag usage. Another feature which is available in industrial grade readers is Wi-Fi availability and programmable reader. This can be taken advantage of and help eliminating
the mobile app entirely. All the necessary code can be programmed into the reader and API calls can be made by connecting to the Wi-Fi. On the other hand, mobile compatible readers can be bought too which are relatively less expensive and bulky and can be used by simply plugging into the phone.

3. **Mobile & Reader Logistics**: Considering the mobile and reader together as a single unit, there is a need to look at how this 'device’ will be handed to and taken from the delivery personnel. A possible solution to this could be usage of a small, dedicated locker system which contains the detection device. This locker is password protected by a dynamic code which is generated by each shop whenever a delivery is to be made. To minimize the hassle and optimize security, the same OTP which was discussed about in the Authentication section above can be used i.e., the delivery personnel obtain the passcode from the shop they’re delivering to, open the locker to obtain the device and authenticate the delivery session. Once the delivery is done, they’re responsible to get it back to said locker and close it up for the next delivery to take place. This proposed solution is contact-less, with no MTR personnel needed to be involved, and since accountability is established by the shop (by generating the OTP) there is minimal risk of theft or damage to the property.

5.6 Received Signal Strength

It is recommended that in the future, to enhance the quality of this system, state of the art RFID readers with received signal strength (RSS) detection be used. This is because the radio frequency signal strength received by these readers can tell us a lot about the spatial information between the tag and the reader. Since it is a challenge to mathematically model the variation of radio frequency signals received in space, methods using statistical analysis are developed to calibrate the relationship between signal strength and distance. Many algorithms have now been developed which can used to better estimate and understand this relationship which will be discussed in this subsection.
5.6.1 Nearest Neighbours

This is perhaps the most intuitive approach. It simply works on, the closer two points are, the smaller the difference between the strength of the received radio frequency signal. As shown in Figure 5.1, if k is set to 4, the x and y coordinate of any point could be easily calculated by the formula

\[
\begin{align*}
    x &= \sum_{i=1}^{k} w_i x_i \\
    y &= \sum_{i=1}^{k} w_i y_i
\end{align*}
\]

Where \((x_i, y_i)\) and \(w_i\) are coordinates and weights of the reference points of the grid and \(k\) is the number of neighbours. The weights or \(w_i\) are calculated by taking out the difference between RF signal strengths between the target tag and the reference point.

Figure 5.1: Schematic of nearest neighbour method [47]
5.6.2 Proximity

As can be viewed in Figure 5.2, the Proximity algorithm, makes use of the approximate communication area o sense whether the ag is present in the region or not. Although this method requires lesser computation, the accuracy achieved is also poorer as a result. Suppose the area to be localized is partitioned into equal sized blocks. Now let detectable area of reader also be divided into equal-sized cells. If the reader detects the tag at m positions within its range, each of these m positions will be a union of cells. The centroid of the intersection of these areas can estimate the location of the tag.

5.6.3 Kernel-based Learning

Kernel-based learning methods estimate the location of tags based on the fact that the smaller the distance between two points in signal space, the smaller the distance is in actual physical space. Instead of transforming signal strength into physical distance by a propagation model, Kernel-based learning models work straight with signal strength. Kernel-based learning itself can be divided in to 2 variations, one is classification and the other is regression. As the name suggests, classification is a little less computationally expensive and estimates whether the tag is in a ‘defined area’. When this is done over multiple overlapping ‘defined areas’, the location of the tag, similar to the Proximity
method can be estimated by finding the centroid of this overlap. Although, this is different from the Proximity method and a bit more accurate since the ‘defined areas’ can take variable shapes.

The second category or regression, is computationally expensive since it directly estimates the value or probability distribution of the coordinates of the tag. This is done through processing the data in three phases, defining the kernel matrix, learning the discriminant function, and online localization [33]. The details of these phases are out of the scope of this project and shall not be discussed.
Chapter 6: Future Work

This project provides an MVP to be tested further. If deemed successful, it can be scaled up and worked upon extensively to add more functionality and finesse to every aspect of the system. The following section describes the expected next steps to be taken for further development.

6.1 Extensive Testing

While this project has carried out tests to show proof of concept, increased vigorous testing is still required to be done to highlight possible flaws in the system. As originally planned, testing phase 1 has been carried out in a minimal disturbance environment to establish proof. The next 2 phases of testing still need to be carried out which are:

1. **Pseudo-realistic MTR-like environment**: Testing in surroundings with moderate to high signal disturbance can help emulate an MTR-like setting. This can help with getting close to real-world data which can in turn optimize calibration of the system. The abundant signal disturbances are expected to yield some data loss or dirty data which convey how robust the system is. The system network can also be optimized if the results are not desirable.

2. **Actual MTR station testing**: Once the MTRC is completely convinced of the performance of the system, before installation it still needs to be tested in the actual environment it needs to function in. This will be done to check for any spatial quirks or inconsistencies any specific location within the station might have. This final phase of testing is expected to be carried out in a single station to see how well the system performs in the actual real-life setting and to find if any final tweaks or changes need to be made before it is installed. Since it has a lot of hardware components which cannot be updated as the need comes and once fitted will be likely used for an extended period of time, thorough testing is an absolute necessity.

6.2 Statistical analysis of Accumulated Data

As part of the original scope of this project, it was expected that there would be enough data accumulated by the end of the testing phase that a statistical tool could be developed to give insight into the shops, their extended deliveries trends and their disciplinary records in terms of violations caused etc. As mentioned in section 3.8, a basic wireframe UI of what was envisioned as the analysis interface has been created as a part of the web app with placeholder data. Due to the time constraint of this project, data analysis simply couldn’t be performed due to lack of accumulated data. With changes to database schemas and inconsistent data from time to time, uniform data could not be
collected. In the future, with successful testing carried out, the database is likely to grow in size and gather large amounts of consistent data to be analysed. The plan laid out had three aims to fulfil:

1. **Display delivery trends:** Graph out how frequently shops within a station get deliveries, what time they usually start, how long they take etc. This will give the MTR the ability to track the needs of different kinds of shops and help them advertise to focused groups in a better way.

2. **Display Violation Stats:** The analysis will show which shops have been getting the highest number of violations in the past day/week/month/year. This can be compared with other stations and the same shops from other locations to identify what the issue is. This can be helpful for the MTR to identify which shops have unacceptable disciplinary records and they might want to warn such shops about the delivery personnel and request them to take necessary action. If needed they may even evict such franchises from the MTR family.

3. **Predictive analysis:** This feature can build upon the trends and use simple regression models to predict upcoming deliveries and delivery times. This can help MTR have better relations with their tenants as well as carry out research with generated predictions.
Chapter 7: Conclusion

The report introduced the risks of in-station delivery to retail outlets in the MTR stations of Hong Kong and provides the basis for the need of an Indoor Positioning and Alert system to track movements of delivery personnel and provide real-time alerts when violations occur.

Extensive research was carried out on existing Indoor Positioning Systems, evaluating the technologies used and determining what would be most suitable for the project based on constraints and requirements. In terms of functionality, the focus of the project was on localization and visualization. RFID technology was identified as the most suitable method of implementation. There is extensive experimentation on RFID localization and on algorithms to counter the major signal strength and reception related drawbacks. It facilitated real-time tracking and is a scalable and cheap solution which can be implemented independent of characteristics of the area to be tracked and has minimal infrastructure requirements.

A robust system consisting of a digital map constructed using RFID tags, a RFID reader with a mobile application interface and a web application functioning as an API and the data analysis and visualization tool was developed. This system was tested in various environments, both indoor and outdoor, with varying levels of interference and noise signals. The system was successful in accurately tracking live position and providing violation alerts, with low latency. To conclude, the objectives of the Senior Design Project were achieved.

The limitations include overhead of generating digital maps of the area to be tracked, and the fact that testing was done with only one entity to be tracked, and the fact that the testing scenarios were approximations of MTR stations.

The development of such a customized Indoor Positioning System requires a firm understanding of the requirements and restrictions. Since there is real-world interaction, which consists of a lot of variables that cannot be simulated, it is of prime importance to conduct various tests of ensure high quality. Direct and simple implementations are highly recommended. Future research can be done on tag deployment strategies, reader signal collisions and RSS algorithms.
Reference List


