



A user's guide

Authors:

Ashley Casper

Joel Elixson

Neil Monday

Rahil Patel

Jeremey Sellen

Table of Contents

Table of Contents	2
Table of Figures	3
Table of Tables	4
1 Introduction.....	5
1.1 Societal Problem	5
1.2 Abstract of Real World Product.....	5
1.3 Introduction to the Prototype	6
1.4 Objectives	7
2 Getting Started	8
2.1 Installation and Configuration of the MySQL Server	8
2.2 Executing the AIR Tracker Application Software.....	12
2.3 Configuring the AIR Tracker Web Interface	12
2.3.1 Installing and Configuring Your Web Server	12
2.3.2 Configuring the AIR Tracker Generator	13
2.4 RuBee Demo Kit.....	14
2.4.1 Hardware Connections	15
2.4.2 RuBee Finder Software Setup.....	17
3 Using the Prototype.....	17
3.1 Graphical User Interface (GUI)	17
3.1.1 Baggage Handler GUI.....	17
3.1.2 Master Baggage Handler GUI	19
3.2 Using the AIR Tracker Generator.....	22
3.3 Using the RuBee system	25
3.4 Using the AIR Tracker Application	27
3.4.1 Creating the Airport	27
3.4.2 Displaying the Routing Matrix	31
3.4.3 Creating Bags.....	31

3.4.4	Viewing Bags in the Check-In Queue	32
3.4.5	Bag Misroutes and Alerts.....	33
3.4.6	Let the Routing Begin!.....	34
4	Error Messages.....	34
4.1	Database Error Messages.....	34
4.2	AIR Tracker Application Error Messages	35
5	Glossary of Terms.....	36

Table of Figures

Figure 1.	AIR Tracker Prototype Major Functional Component Diagram.....	7
Figure 2.	Successful Creation of the Airport and AIR Tracker Databases	11
Figure 3.	AIR Tracker Generator Script - http://localhost/airtracker/populate.php	13
Figure 4.	RuBee Demonstration Kit Components	15
Figure 5.	Hardware Connection Setup.....	16
Figure 6.	Baggage Handler GUI	18
Figure 7.	Baggage Handler GUI Non-Alert Screen.....	19
Figure 8.	Editing the Master Baggage Handler Alert with OpenRPT	20
Figure 9.	Example Output of Master Baggage Handler Alert	20
Figure 10.	Master Baggage Handler Test Harness	21
Figure 11.	Master Baggage Handler Alert Screen	22
Figure 12.	Successfully Created Simulation Data	23
Figure 13.	Airport Database Assets	23
Figure 14.	AIR Tracker Database Assets.....	24
Figure 15.	Luggage Carts Containing Bags.....	25

Figure 16. RuBee Finder Software Icon	26
Figure 17. RuBee Finder Software Interface	27
Figure 18. Main menu screen.....	28
Figure 19. "Create Airport" menu screen.....	29
Figure 20. Error Message During Data Generation	34

Table of Tables

Table 1. Causes of AIR Tracker Application Errors.....	35
--	----

[This space is intentionally left blank]

1 Introduction

1.1 Societal Problem

When an airline does not know where a traveler's bag is and that bag is never given back to the traveler, the bag is considered lost. When a bag is not readily available when a traveler lands, the bag is considered mishandled. Lost and mishandled luggage is a problem that has always plagued the airline industry, and this problem costs the airline industry 3.8 billion dollars a year. This number covers all facets of lost luggage, including payments to cover permanently lost luggage, home delivery costs, and rerouting costs. For domestic airlines, about 5.26 pieces of luggage are lost out of every 1000 (U.S. Department of Transportation, 2008). Globally, the average is much greater at 19 mishandled pieces of luggage per 1000. There have been many attempts to solve this problem, but since 2002, this problem is still getting worse (SITA, 2009).

1.2 Abstract of Real World Product

AIR Tracker plans to tackle this lost luggage problem on many different fronts. First, there will be a new Radio Frequency Identification (RFID) reader added to all luggage carts in all airports that implement the AIR Tracker system. Additionally, an RFID tag will be attached to all bags that are routed through the system. This module will constantly scan all bags onboard carts in order to gain a higher resolution of a bag's location at any given point in the routing process.

Secondly, AIR Tracker will improve upon the airport's belt routing system by using RFID scanners to replace the inefficient barcode scanners. Barcodes have many pitfalls that make them less than sufficient for luggage routing. Tags with barcodes that

have been smudged, folded, or damaged in any way significantly reduce the barcode scanner's ability to read the tags. Alternatively, RFID tags do not suffer from the same problems because they use radio waves and do not need a clear line of sight.

Also, AIR Tracker will provide airports the ability to see historical data on their baggage routing process. This reporting will allow airports to pinpoint areas of the baggage routing process that may not be working efficiently.

1.3 Introduction to the Prototype

The AIR Tracker prototype is based upon a medley of different technologies that work together to show basic luggage tracking and functionality. You can use the prototype to show the actual reading and processing of RuBee tags using the RuBee reader. The prototype can also be used to demonstrate a simulated airport routing system with AIR Tracker's alerting and routing capabilities. The laptop you decide to use will run two databases; those databases include a simulated airport database and an AIR Tracker database. The prototype will also be capable of displaying simulated handheld alerts. Figure 1 shows the major functional components of the prototype.

[This space is intentionally left blank]

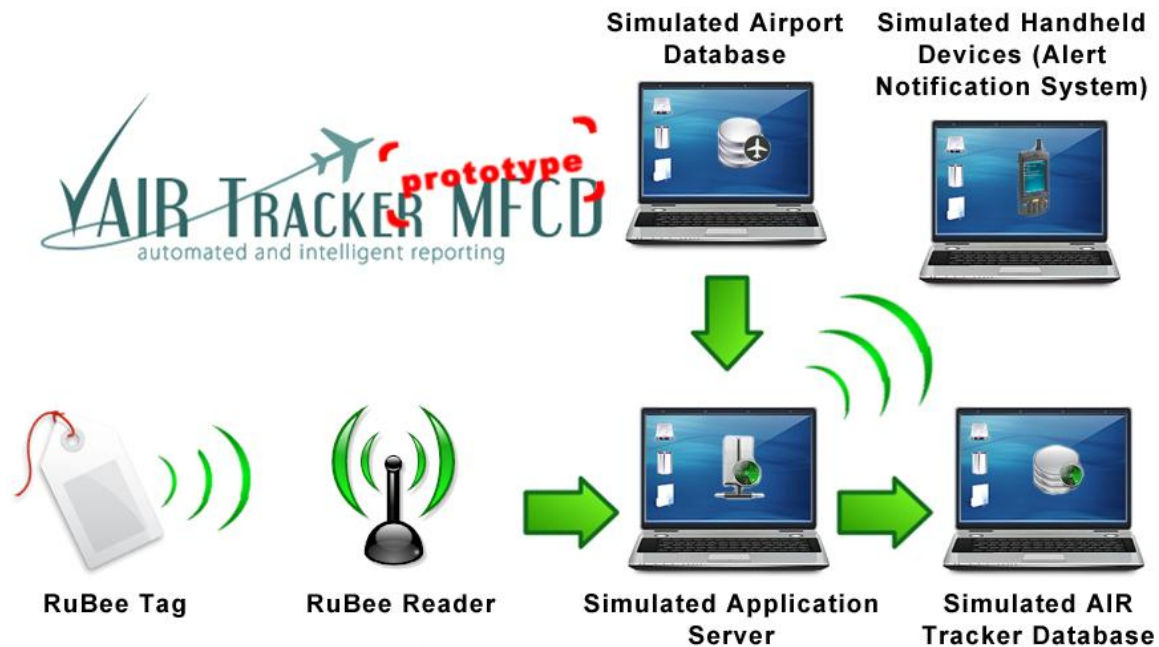


Figure 1. AIR Tracker Prototype Major Functional Component Diagram

1.4 Objectives

The main goal of the prototype is to show that it is feasible to build a real-world product that could realistically help solve the lost luggage problem. By using the prototype, you will be able to see that the main aspects of the AIR Tracker system are feasible. The prototype will show that:

1. physical tag scanning is practical for a real world situation
2. alerts can be captured and stored during any point in the luggage routing process
3. real-time alerts can be triggered to warn baggage handlers, and
4. it is possible to generate reports accounting for all historical alerts.

[This space is intentionally left blank]

2 Getting Started

Before you begin using the AIR Tracker prototype, you must first configure your local machine. It is strongly recommended that you use a Unix-based operating system for all software installations and configurations. Configuration relies heavily on the g++ compiler and the Apache web serving software, which are standard components of Unix-based operating systems. All modern operating systems include some form of command-line interface. You'll need to locate your command or terminal application and leave it open until you've reached Section 2.2.

2.1 Installation and Configuration of the MySQL Server

The AIR Tracker prototype's core application relies heavily on information storage and retrieval; therefore, before you begin using the prototype, you must first install and configure the MySQL Community Server software. It is recommended that you download MySQL Community Server version 5.1; however, version 5.4 is also capable of performing all database related tasks required by the prototype. Sun Microsystems have done a fantastic job of making their MySQL software available for every major platform. MySQL software can be downloaded, free of charge, from the following URL:

<http://dev.mysql.com/downloads/mysql/5.1.html> - downloads

Installation of Sun's MySQL software is different for every platform; therefore, instructions on installing MySQL will not be covered in this document. You may refer to

the MySQL version 5.1 Reference Manual for installation instructions related to your particular operating system. Installation or upgrade instructions can be viewed using the URL below:

<http://dev.mysql.com/doc/refman/5.1/en/installing.html>

Now that you've installed your MySQL server software, let's get to work. Using a command-line shell or terminal application will allow you to connect to your newly installed (or upgraded) database. We'll use the command-line to create a new root user and to grant the root user full authority over the '**airport**' and '**airtracker**' databases.

Open your command-line terminal application. We'll begin by creating a root user account. This allows you to administer your MySQL server. In your terminal window, type the following command (not including the % sign):

```
% mysqladmin -u root password NEWPASSWORD
```

When prompted to enter a password, you may enter a password of your choosing. Keep your terminal window open; you'll need it later.

Now, let's download the MySQL database file needed to create the database schemas required by the AIR Tracker prototype. The SQL file can be downloaded from:

<http://www.neilmonday.com/airtracker/prototype.htm>

Next, we'll need to load the '**airport**' and '**airtracker**' database schemas from the

'airtracker.sql' file you just downloaded. In order to do this, you'll need to know the path of the file. Fortunately for Unix users, you can simply drag and drop the file into your terminal window after typing the following command (but before you press return):

```
% mysql -h localhost -u root -p <
```

The absolute path of the **'airtracker.sql'** file will appear next to the input character (<). For Windows users, it is recommended that you place the **'airtracker.sql'** file into your C:\ directory (or equivalent):

```
% mysql -h localhost -u root -p < C:\airtracker.sql
```

Press the return button on your keyboard and input your **'root'** user password as prompted. Now, the database schemas used by the AIR Tracker prototype have been installed.

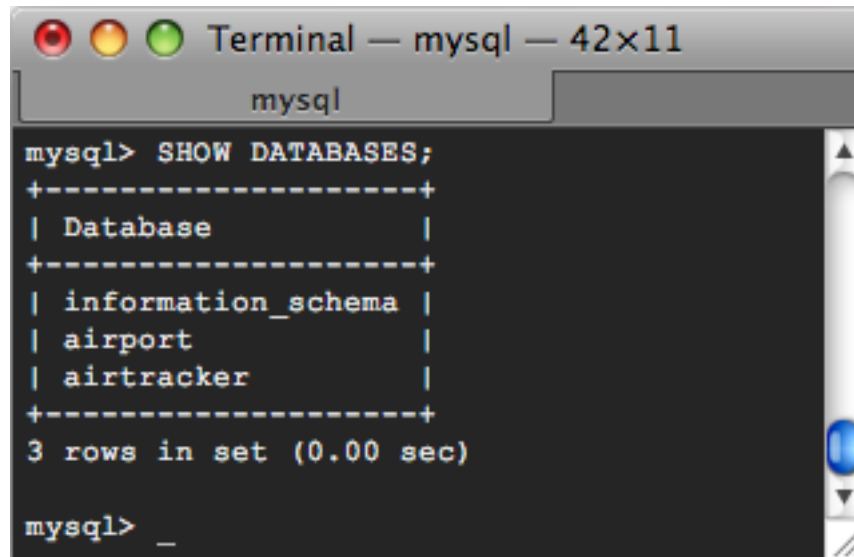
Let's take a look! First, let's log in by typing the following command:

```
% mysql -h localhost -u root -p
```

Enter your password when prompted. Next, type the following command (not including the **'mysql>'** prompt):

```
mysql> SHOW DATABASES;
```

You should see something similar to Figure 2.

A screenshot of a macOS Terminal window titled "Terminal — mysql — 42x11". The window shows a MySQL command prompt where the user has entered "SHOW DATABASES;". The output is a table with one column "Database" and three rows: "information_schema", "airport", and "airtracker". The prompt "mysql> _" is visible at the bottom.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| airport      |
| airtracker   |
+-----+
3 rows in set (0.00 sec)

mysql> _
```

Figure 2. Successful Creation of the Airport and AIR Tracker Databases

Next, we're going to create a new user (the one used by the AIR Tracker prototype application) and grant that user the appropriate permissions; executing the following commands can do this:

```
mysql> CREATE USER guest;

mysql> GRANT ALL PRIVILEGES ON airport.* TO 'guest'@'localhost';

mysql> GRANT ALL PRIVILEGES ON airtracker.* TO 'guest'@'localhost';
```

This creates a new user 'guest' without a password. Don't worry! The user only has the ability to interact with the '**airport**' and '**airtracker**' databases and cannot affect any other database you may have already created.

That's it! You've successfully installed and configured the MySQL Community Server software and created all the necessary databases schemas! Next, we'll discuss how to obtain and run the AIR Tracker application.

2.2 Executing the AIR Tracker Application Software

The AIR Tracker application comes packaged as a Unix executable; however, you may download the source code as well. The AIR Tracker application can be found here:

<http://www.neilmonday.com/airtracker/prototype.htm>

Compiling the AIR Tracker source code on a non-Unix platform requires a C++ compiler, MySQL++ source files, and significant changes to the AIR Tracker application source code. These modifications to the AIR Tracker prototype are not covered in this manual. You must utilize external resources to discover how to properly integrate MySQL++ with a C++ coding environment.

2.3 Configuring the AIR Tracker Web Interface

The AIR Tracker prototype requires the use of web server software. More specifically, it requires a web server hosted on your local machine that is capable of handling and visualizing PHP scripts. AIR Tracker's prototype uses PHP to generate simulation data. Use of the Generator scripts is discussed in Section 3.2. If you already have a web server installed, you may skip to Section 2.3.2.

2.3.1 Installing and Configuring Your Web Server

Installing, configuring, and running your web server is not discussed in this manual; however, the Apache website contains all the necessary documentation to

perform such actions. You may obtain the latest version of the Apache web server from the following URL:

<http://httpd.apache.org/download.cgi>

Instructions on installing and running your web server can be obtained by using the link below:

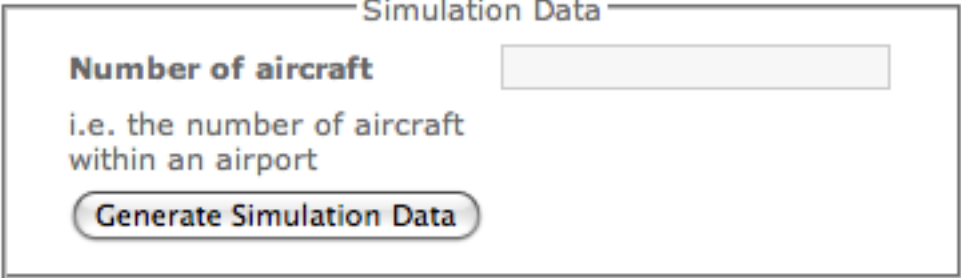
<http://httpd.apache.org/docs/2.2/>

2.3.2 Configuring the AIR Tracker Generator

In order to populate the Airport and AIR Tracker databases, you must first download the Generator PHP scripts and **install them within a subdirectory of your web server**¹. The Generator scripts can be downloaded from:

<http://www.neilmonday.com/airtracker/prototype.htm>

The directory that hosts your web server is operating system specific; refer to the Apache documentation for questions concerning its location. Once you've successfully installed the AIR Tracker Generator scripts, the script URL below should resemble Figure 3 when viewed in your web browser.



Simulation Data

Number of aircraft

i.e. the number of aircraft within an airport

Generate Simulation Data

Figure 3. AIR Tracker Generator Script - <http://localhost/airtracker/populate.php>

That's all! You're all set up! Now, proceed to Section 2.4 to find out how to prepare the hardware portion of the prototype.

¹ You can put the Generator scripts within a subdirectory of your choice. The above URL uses a subdirectory called 'airtracker', for example, but you can put the scripts anywhere within your web server directory, even within your root directory!

2.4 RuBee Demo Kit

The major hardware and software components of the RuBee Demo Kit are illustrated in Figure 4 and are as follows:

1. Receiver
2. Ranger Antenna
3. Four RuBee Radio Tags
4. CD-ROM with RuBee Finder Software & Demo Kit Manual
5. Associated Cables and Power Supplies

[This space is intentionally left blank]

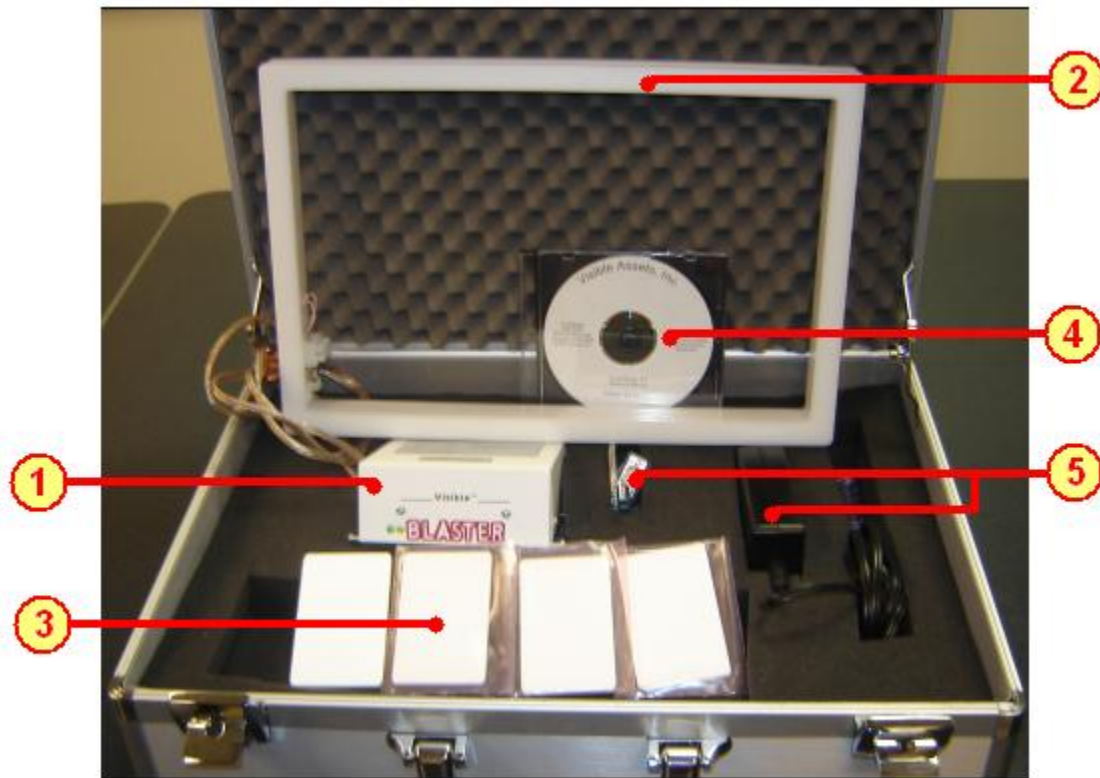


Figure 4. RuBee Demonstration Kit Components

A laptop will be required for the RuBee hardware setup and software installation. The laptop should include a USB or a RS 232 serial port and an operating system of Windows XP or Vista.

2.4.1 Hardware Connections

The hardware of the RuBee Demo Kit should be connected to the laptop as shown in Figure 5.

[This space is intentionally left blank]

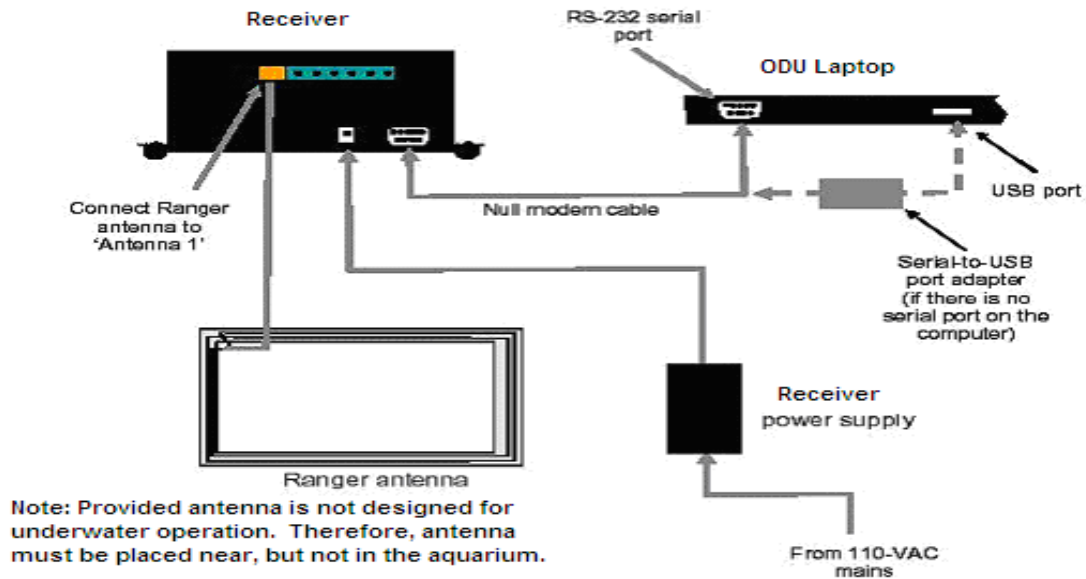


Figure 5. Hardware Connection Setup

1. Attach the antenna to the receiver with provided cable.
2. The receiver is connected to the laptop's RS-232 serial port using a null modem cable.
3. If there is not a serial port on the laptop, the null modem cable will be connected from the receiver to the serial-to-USB port adapter, then to the USB port on the laptop.
4. The receiver is connected to a power supply.
5. (See the Demo Kit Manual for more hardware setup instructions and troubleshooting)

2.4.2 RuBee Finder Software Setup

The RuBee Finder Software CD, which is included in the RuBee Demo Kit, needs to be installed on a laptop. Use the following steps to install the Finder Software on your computer:

1. Insert the Finder CD into your computer CD-ROM drive.
2. On your computer click the 'Start button', then select 'Run'.
3. Click the 'Browse' button and find your CD-ROM drive.
4. Double click on the CD-ROM drive in the Browser window.
5. Double click on 'Finder Setup'.
6. Press OK in the 'RUN' window to begin installing the Finder Software.
7. Click 'Next' 4 times
8. Click Install
9. After the software has been installed, click 'Finish'

Congratulations! The hardware portion is now done and you are ready to get into section 3, which encompasses using the prototype.

3 Using the Prototype

3.1 Graphical User Interface (GUI)

3.1.1 Baggage Handler GUI

The Baggage Handler GUI is a simple and easy-to-use interface that will prove AIR Tracker's ability to keep a record of all bags being routed through an airport's system. Figure 6 shows this GUI.

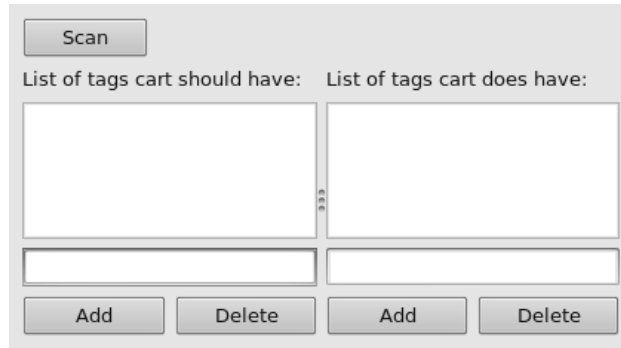


Figure 6. Baggage Handler GUI

The left pane is the “should have” pane, listing those bags which a cart or scanner should have according to the itineraries of the bags. The right pane is the “does have” pane, listing those bags that a cart or scanner “physically” has at that particular moment.

Underneath both panes, you’ll see two buttons, an “Add” button and a “Delete” button, designated for each listing.

To add a bag ID, simply type the ID into the text field just below the desire pane and click the “Add” button.

To delete a bag ID, select the ID from the desire list by clicking on the ID number and clicking the “Delete” button.

To determine whether the alert system is working properly, enter two identical bag IDs into both the “should have” and “does have” panes.

To do this, type an ID, such as “123,” into the text field just below the left “should have” pane and click the “Add” button just below the text field.

Enter the same ID into the text field just below the right “does have” pane and click the “Add” button just below the text field. You should see that both panes have the bag ID listed as in Figure 7.

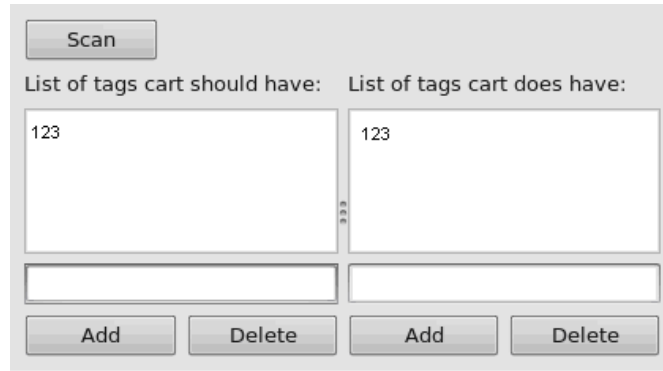


Figure 7. Baggage Handler GUI Non-Alert Screen

Finally, click the “Scan” button located at the top-left of the Baggage Handler GUI. Since both bag IDs are listed as belonging to the cart or scanner and being on the cart of scanner, you’ll receive no errors and know that the AIR Tracker prototype is correctly tracking all bags registered within the system.

3.1.2 Master Baggage Handler GUI

The Master Baggage Handler will have three GUI tools to use. There is the OpenRPT report writer, the Baggage Handler GUI and the Master Baggage Handler Alert Screen. OpenRPT is an open source tool used to write SQL reports that can be neatly formatted for easy reading. OpenRPT is developed by the company xTuple and can be downloaded from SourceForge.net. AIR Tracker comes with an example of an alert report, but you can create and design your own custom reports with OpenRPT. This tool has a WYSIWYG (What You See Is What You Get) editor that you can use to format your reports to your liking. Figure 8 shows the editing interface for the OpenRPT software.

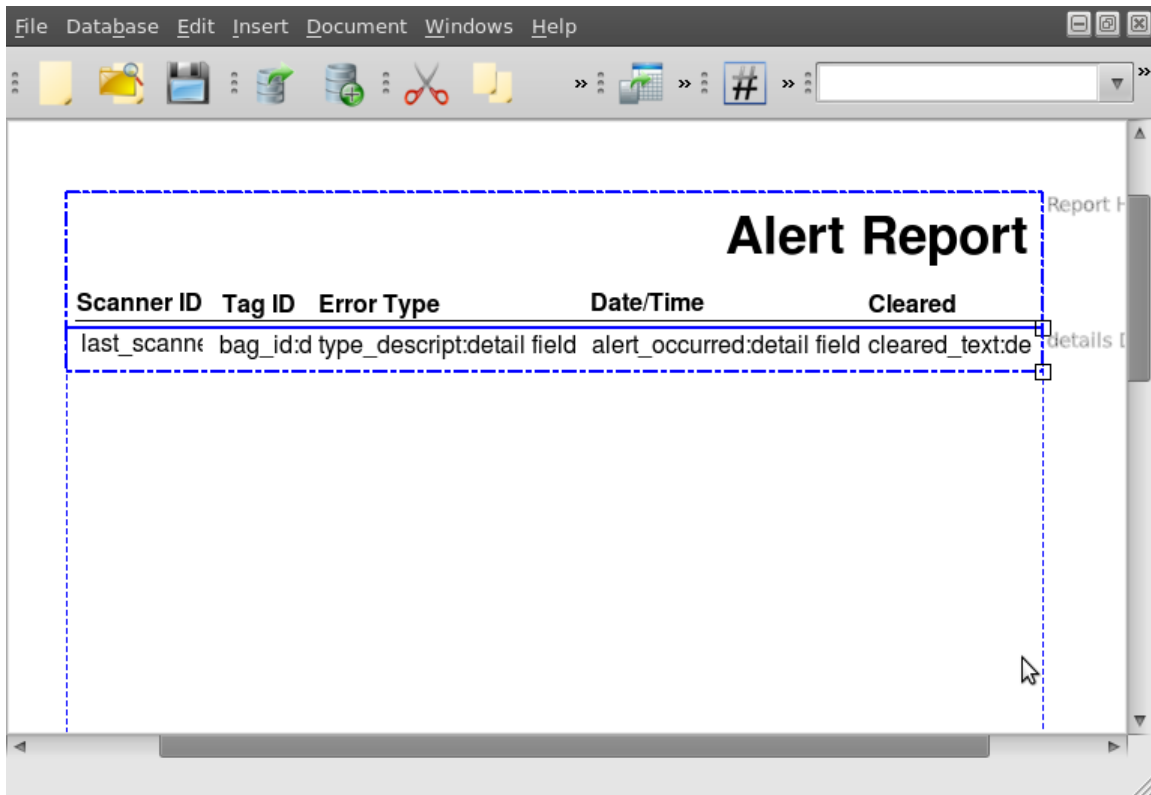


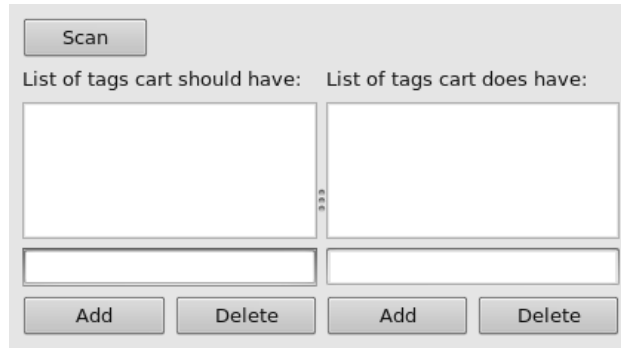
Figure 8. Editing the Master Baggage Handler Alert with OpenRPT

Figure 9 shows the actual output once the report is completed.

Alert Report				
Scanner ID	Tag ID	Error Type	Date/Time	Cleared
1000	1	Luggage cart scanner	2009-04-22T13:37:28	No
1001	2	Belt loader scanner	2009-04-22T13:37:28	Automatic

Figure 9. Example Output of Master Baggage Handler Alert

Just like the Baggage Handler, the Master Baggage Handler will have the GUI displayed in Figure 10 at his disposal.



The image shows a software interface titled "Master Baggage Handler Test Harness". At the top left is a "Scan" button. Below it are two labels: "List of tags cart should have:" and "List of tags cart does have:". Each label is followed by a large, empty rectangular box for a list of tags. To the right of these two boxes is a vertical scrollbar. Below each list box is a smaller, empty rectangular text input box. At the bottom of the interface are four buttons: "Add" and "Delete" under the first list, and "Add" and "Delete" under the second list.

Figure 10. Master Baggage Handler Test Harness

You can add tags using the text boxes and clicking the “Add” button. You may also delete tags from the two lists by highlighting the tag and clicking on the “Delete” button. This can be done for both the “should have” and “does have” lists. Once you have added the tags that you would like to be in the test harness, you may click the “Scan” button to process the tags. If there are any inconsistencies in the two lists, an alert will be generated. Figure 11 displays the error screen you will get when an inconsistency between the two windows exists.

[This space is intentionally left blank]



Figure 11. Master Baggage Handler Alert Screen

This GUI will present a dialog box showing all alerts. There will be two lists of tags where each tag can be manually cleared using the buttons at the bottom of the dialog box.

3.2 *Using the AIR Tracker Generator*

The AIR Tracker generator is fairly easy to use, and understanding the visualized data is just as easy. The Generator creates virtual assets for use in the airport routing simulations created by the AIR Tracker application. To generate new simulation data, simply run the **‘populate.php’** script, input the preferred number of aircraft, and press the **‘Generate Simulation Data’** button. If you input a number between one and ten, you should see a notification of success similar to Figure 12.

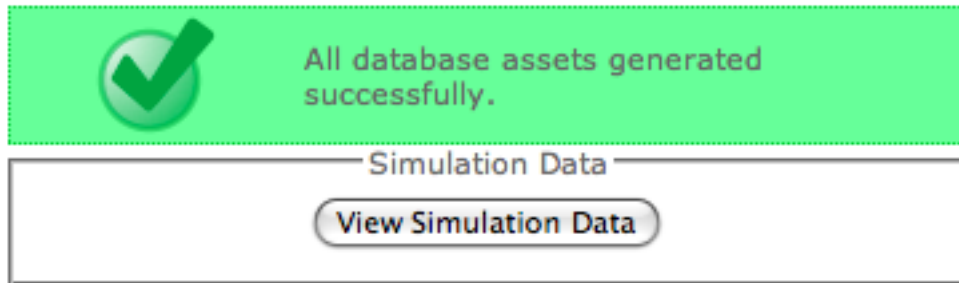


Figure 12. Successfully Created Simulation Data

Next, you may select the **‘View Simulation Data’** button to view the generated assets.

The Airport DB screen, shown in Figure 13, displays all entries in the Traveler table, the Bag table, and the Flight table from the **‘airport’** database. From this screen, you may proceed to generate new simulation data, or view the generated entries within the AIR Tracker database.

Airport DB								
View AIR Tracker Database				Generate New Data				
TravelerID	First Name	Last Name	FlightID	BagID	TravelerID	FlightID	AirplaneID	Gate_No.
1	Arthur	Adams	CO2772	1	1	JW2025	N2747F	9
2	Angus	Allen	EU1021	2	1	EE1197	N1454S	8
3	Aiken	Anderson	PW4678	3	2	RQ2215	N6999X	7
4	Boris	Bailey	VT3089	4	3	VT3089	N3206D	6
5	Arnold	Baker	NG3557	5	3	PW4678	N2401Z	5
6	Cary	Barnes	JW2025	6	4	CO2772	N6362A	4
7	Bowen	Bell	VT3089	7	5	NG3557	N7784T	3
8	Brooke	Bennett	NG3557	8	7	EU1021	N4978R	2
9	Broderick	Brooks	RQ2215	9	7	TL1155	N2755E	1
10	Abner	Brown	TL1155	10	8			
11	Carver	Butler	TL1155	11	8			
12	Aubrey	Campbell	EU1021	12	10			
13	Axel	Carter	JW2025	13	11			
14	Alston	Clark	TL1155	14	12			
15	Barrett	Collins	VT3089	15	12			
16	Bert	Cook	RQ2215	16	13			
17	Blake	Cooper	EE1197	17	16			

Figure 13. Airport Database Assets

The AIR Tracker database should appear similar to Figure 14. The AIR Tracker DB screen displays all entries in the Bag table, Carrier table, Cart table, and Loader table from the AIR Tracker database.

AIR Tracker DB

[View Airport Database](#) [Generate New Data](#) [View Bags on Carts](#)

BagID	FlightID
1	CO2772
2	CO2772
3	EU1021
4	PW4678
5	PW4678
6	VT3089
7	NG3557
8	VT3089
9	VT3089
10	NG3557
11	NG3557
12	TL1155
13	TL1155
14	EU1021
15	EU1021

BagID	CartID
1	4
2	4
3	2
4	5
5	5
6	6
7	3
8	6
9	6
10	3
11	3
12	1
13	1
14	2
15	2

CartID	ScannerID	Gate_No.
9	39	9
8	38	8
7	37	7
6	36	6
5	35	5
4	34	4
3	33	3
2	32	2
1	31	1

LoaderID	ScannerID	Gate_No.
9	49	9
8	48	8
7	47	7
6	46	6
5	45	5
4	44	4
3	43	3
2	42	2
1	41	1

Figure 14. AIR Tracker Database Assets

Again, you may choose to generate new simulation data. Alternatively, you may navigate backward to the Airport DB screen, or view a breakdown of which luggage carts contain which bags. Similar options are available within the luggage cart breakdown screen. Figure 15 illustrates the luggage cart breakdown view.

[This space is intentionally left blank]

AIR Tracker DB					
View AIR Tracker Database			Generate New Data		
CartID	BagID	CartID	BagID	CartID	BagID
1	12	2	3	3	7
1	13	2	14	3	10
1	55	2	15	3	11
1	56	2	39	3	20
1	85	2	40	3	21
1	86	2	48	3	22
1	90	2	49	3	27
		2	64	3	28
		2	65	3	29
		2	66	3	37
		2	67	3	57
		2	87	3	68
		2	95	3	98
		2	99		
		2	100		
				4	1
				4	2
				4	30
				4	31
				4	60
				4	61
				4	72
				4	73
				4	96
				4	97
				5	4
				5	5
				5	23
				5	24
				5	35
				5	38
				5	44
				5	45
				5	46
				5	47
				5	51
				5	52
				5	53

Figure 15. Luggage Carts Containing Bags

Now let us examine how to use the software that came with the RuBee hardware.

3.3 Using the RuBee system

Before you proceed, make sure you have installed the RuBee Finder Software and connected the hardware components according to section 2.4. Follow the below steps to use the RuBee Finder Software:

1. Locate and double-click the Finder software shortcut on your desktop depicted in.

Figure 16)

[This space is intentionally left blank]

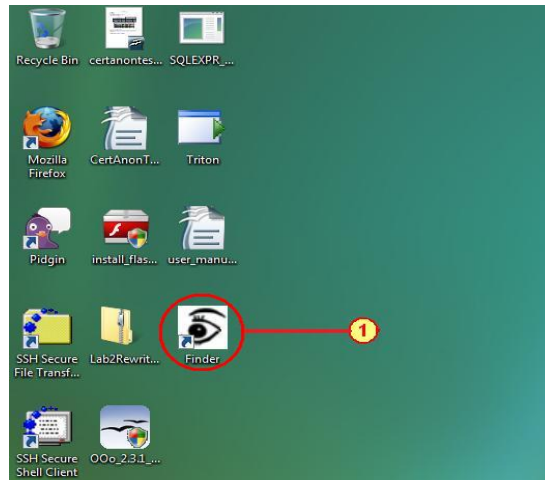


Figure 16. RuBee Finder Software Icon

2. Select the proper “com” from the communication port drop down list.
 3. Select Discovery Mode in order to find any tags that are within range.
 4. After all tags are recognized by the software, select List Mode to place the software in the normal operation mode.
 5. Tag IDs within range will turn green. Tag IDs not within range will turn red.
- (Note: Antenna’s range is 6 - 12 feet depending on the environment).

Figure 17 shows the screen you interface with the RuBee software.

[This space is intentionally left blank]

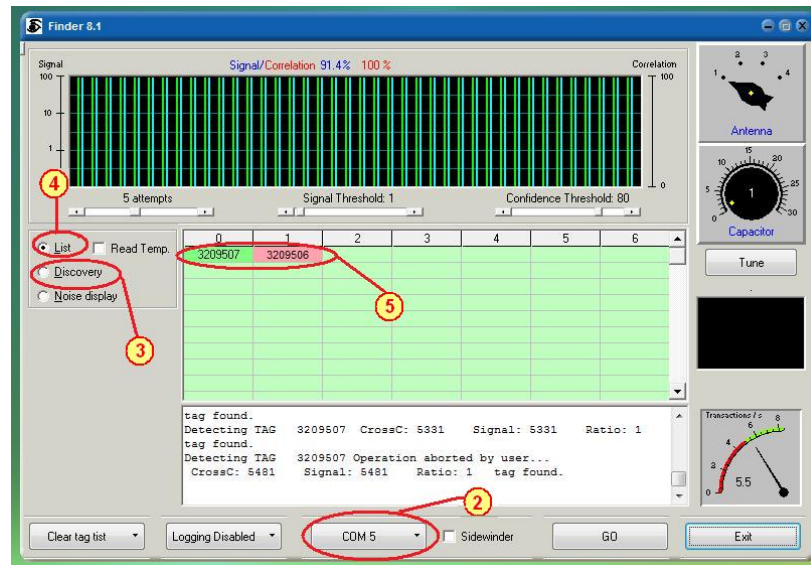


Figure 17. RuBee Finder Software Interface

For information regarding the inner workings of this interface, see the RuBee Kit Manual found at <http://www.rubee.com/DemoKit/>.

3.4 Using the AIR Tracker Application

In very powerful cars, you have very powerful engines driving them. Such is the case with this prototype. The AIR Tracker application at its heart yearns to please the avid user. So sit back, relax and follow along to discover how to properly operate this 6-cylinder beast.

3.4.1 Creating the Airport

So, you've made it this far successfully. You've connected the hardware, installed and populated the databases and are now ready to begin the fun part; actually running the

application. This application comes with many features. Most of these features are accessible from the main menu window (displayed in Figure 18).

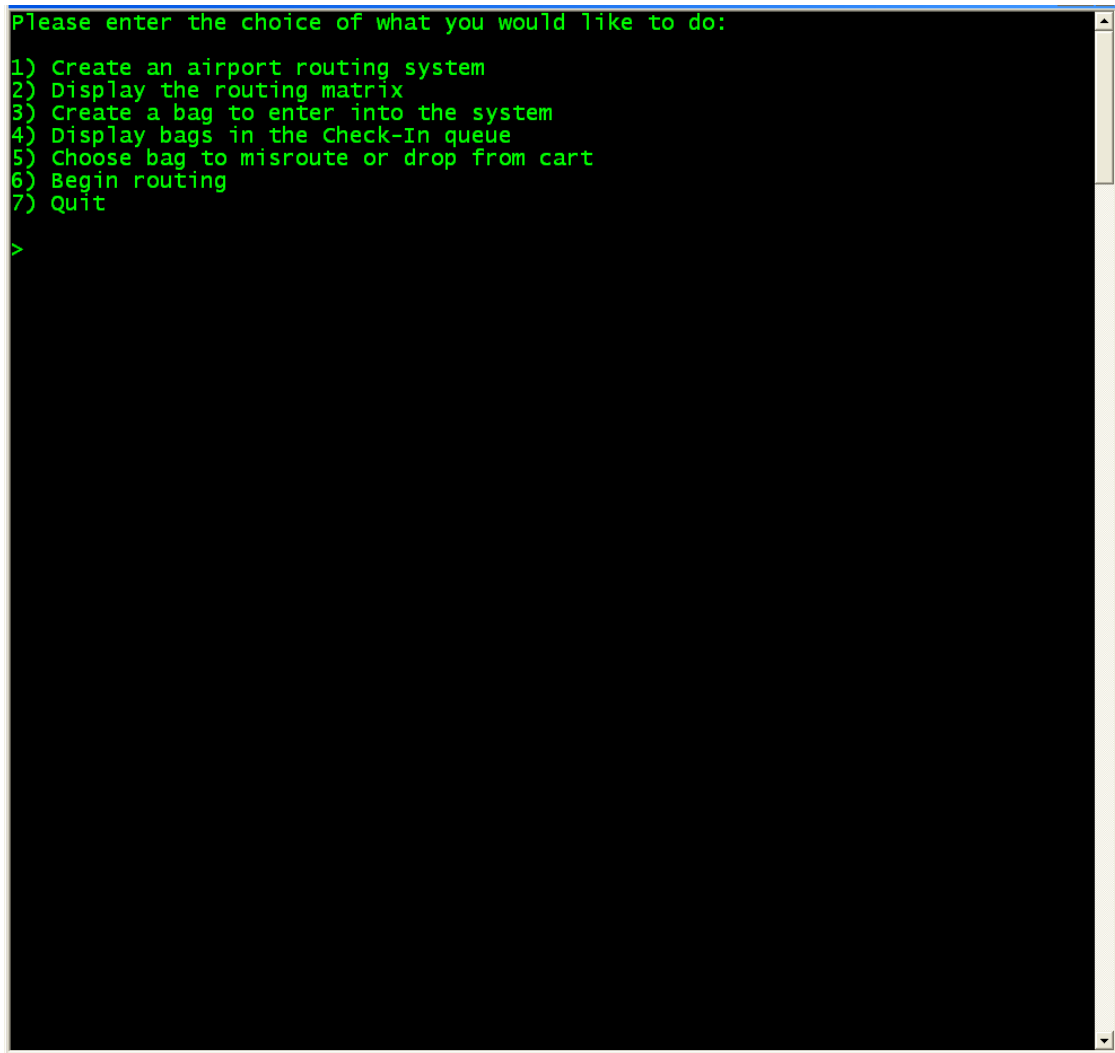


Figure 18. Main menu screen

Selecting menu option [1] from the main menu allows you to create an airport routing architecture and presents you with the screen shown in Figure 19.

```
Do you want to create the airport:
1) Automatically from file
2) Manually
>
```

Figure 19. "Create Airport" menu screen

The menu comes with two airport creation methods - from a file or manual creation. If you choose menu option [1] from this "Create Airport" submenu, the routing system will pull data from a text file and create a luggage routing architecture based on the number of aircraft you previously specified. For example, if you choose to generate simulation data with nine aircraft, the resulting routing architecture will contain nine gates and nine luggage carts toward which each bag may be routed. After the application successfully loads the appropriate text file, you will be greeted with the message

"Airport Routing System Created Successfully!"

If option [2] is chosen, then the system will ask you to enter the number of gates and scanners that will be used in creating the routing matrix for this particular simulation.

The system will prompt you to:

Please enter number of gates:

At this prompt, you must enter an integer from 2 to 10. (It wouldn't make much sense to have zero or one gates, as such an architecture is not much of a challenge for a routing system.) Upon pressing Enter, you will be asked:

Please enter number of scanners:

At this prompt, you must enter an integer from 3 to 100.¹

¹Remember: you will have to manually enter a complete set of valid scanner-to-gate permutations in the next step.

Once you have manually entered a valid number of gates and scanners, the system will begin the process of building the routing matrix. You will be greeted with the following text:

Please enter the valid scanners for each gate. Separate each entry by a space. Enter 0 as the last value and press <ENTER>.

Since zero is not a valid gate number, it will serve as the delimiter between each set of scanners and their corresponding gate. Be sure to separate the scanner numbers by a space and not a comma, semicolon, or something else as equally pleasing.

After you have entered a zero at the end of the line for the first gate, the application takes your input and creates the first row of the routing matrix. The application will now repeat this process for however many gates you told the computer to create.

Congratulations, you are now solely responsible for creating the routing matrix that will serve to be the governing agent in routing all of the virtual tags within the

system! However, the proof, as they say, is in the pudding. So, let's break out the snacks and see what you've done, shall we?

3.4.2 Displaying the Routing Matrix

Now that you have created the routing matrix, option [2] from the main menu allows you to view it. Upon choosing this option, something similar to the following information should be displayed:

```
Gate#1 Scanners: <1 4>
Gate#2 Scanners: <1 3 12>
Gate#3 Scanners: <1 2 8>
```

The above example shows each gate number followed by a list of the scanners that lead to that gate in their respective order.

3.4.3 Creating Bags

Now that you have created the routing matrix and have viewed the contents, the application hungers for something to route. It's now time to enter bags into the routing system you have created. You have the ability to create bags with option [3] from the main menu screen. The system will prompt you to:

- 1) Load from the AIR Tracker database?
- 2) Manually create?

If you choose to load from the database, the application retrieves the luggage data from the AIR Tracker database and loads the check-in queue with a list of bags along with their destination gates. If, on the other hand, you choose to enter the bag information manually, simply enter the bag ID number and the destination gate when prompted to do so.

3.4.4 Viewing Bags in the Check-In Queue

After creating any number of bags, you can view them by choosing option [4] from the main menu screen. This will display all of the bag identification numbers and destination gates that you have assigned thus far. The format of the output is as follows:

```
BagID: 845
Destination Gate: 7

BagID: 849
Destination Gate: 10
```

At this point, you've created the routing matrix and have also created the bags to route through the system. Now there are two primary options afforded to you. You can choose option [6] and begin routing, or you can decide to manipulate the system a bit with option [5].

Choosing option [6] will route all of the bags you have created through the routing matrix just as it should, with no errors. Upon completion of the routing you will see:

All bags have routed!

This only indicates that all bags have been successfully routed. No bags have flagged an alert. To be more useful, let's stir the waters a little more and choose option [5].

3.4.5 Bag Misroutes and Alerts

If you would like to force an alert to flag, you can choose option [5] and you will be asked to choose the following:

Would you like the bag to:

- 1) Misroute in the internal routing
 - 2) Drop from a cart in the external routing

Choosing option [1] from this menu will prompt you to enter the bag number to be affected (i.e. the bag you want to misroute). Enter a bag number that you have already created.²

²If you enter a bag number you have not created, you will get the error stating that the bag has not been created yet, and you will be returned to the main menu.

Choosing option [2] from this menu also prompts you to enter a bag number to be affected. In this instance, the bag will route properly to its correct loading bay, but it will be dropped from its correct cart. The same error will flag as the misroute option if you enter an invalid bag number.

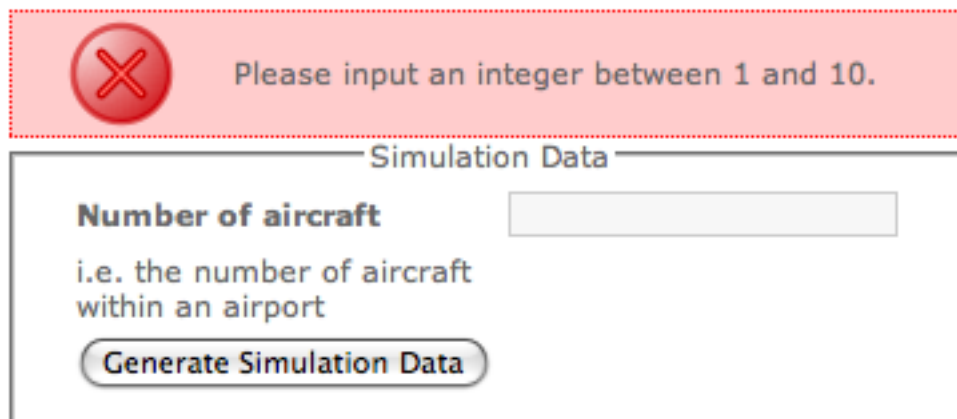
3.4.6 Let the Routing Begin!

Now, the only thing left to do is to select option [6] and let the application do its thing. You will see the application chewing away at all the bags and the multiple paths via the scrolling log interface. Once the process is done and you are greeted by the following message “All bags have routed!” and you can access the alerts.txt file to see which bags triggered an alert and where each alert was triggered. Additionally, you can access the information through the database PHP interface.

4 Error Messages

4.1 Database Error Messages

When generating new simulation data as discussed in Section 3, you may have accidentally input a non-numerical character, or you may have input a number of aircraft not within the one to ten range. In this case you should see an error message similar to Figure 20. Don’t panic! Simply input the appropriate number to continue.



The image shows a web application interface. At the top, there is a red error message box with a red 'X' icon and the text "Please input an integer between 1 and 10." Below this is a form titled "Simulation Data". The form contains a label "Number of aircraft" followed by a text input field. Below the input field, there is a smaller text label "i.e. the number of aircraft within an airport". At the bottom of the form is a button labeled "Generate Simulation Data".

Figure 20. Error Message During Data Generation

4.2 AIR Tracker Application Error Messages

Since we are human, sometimes we need to be told that we are wrong. AIR Tracker provides no escape from that stark reality. Table 1 displays the error messages that may be encountered while using the application as well as the cause.

Error	This error is received if...
I'm sorry, but you first need to create an airport.	you attempt to display the routing matrix before you have created it.
I'm sorry, but you have to create bags first.	you attempt to choose a bag to misroute before you have created any bags.
I'm sorry, but that bag has not yet been created.	you attempt to choose a bag to misroute that has not been created.
I'm sorry, but that is an invalid answer, please try again.	you enter an invalid input for a question from the system.

Table 1. Causes of AIR Tracker Application Errors

[This space is intentionally left blank]

5 Glossary of Terms

GUI – Graphical User Interface: The human helpful method that is used to interact with a software module.

MySQL: Database software used in conjunction with the AIR Tracker application.

RFID-Radio Frequency IDentification: Wireless method of communicating between two nodes.