**Janik Singh 1159963**

**Functions**

* Give new bin destination/cities
* scan package check it can go in given bin
* scan bin whenever it moves and apply scan to all contents

**Uses Cases**

**assign destination city**

The Package sorter indicates that he would like to create a new bin whilst giving it a destination that it will be delivered to. He specifies a destination city and creates a new bin, where packaged with the same destination will eventually be put it.

**scan package to go in bin**

The package sorter indicates that he would like to scan packages to check if it can go in given bin. If the package destination and bin destination are not compatible, the system will notify the package sorter that the package cannot be placed in that bin. Otherwise the system will confirm the package and bin are compatible and display a message showing that it has been accepted. If that package already exists, then the system will notify this and assume it was scanned accidently the second time. If the bin has exceeded its package limit, then the system will notify this for a new bin.

**scan bin when it moves**

The package sorter indicates that he would like to scan a bin whenever it enters and leaves the facility. When the bin is scanned to move, the scanning information is also applied to each package in it.

**Object Model Diagram**



**Class Diagram**



**Testing Plan**

* Make sure a new bin is created with a destination city defined
* Check the destination of package and bin to see if it functioned properly and have been saved
* Make sure scan history of bin and all packages in it have been changed successfully and have been saved

**Missing requirements**

Scanning device: used for scaning Packages and bins

Database: we just another java file to test inputs to make sure they work, a database would have been easier to test with.

**Process**

We started by reading through the Group Assignment. At our first group meeting, We discussed the Assignment and pondered the specifics and what the end goal might look like. We identifed the functions of the system. Initially we had 16 functions, but had not yet decided who would do what just yet. We agreed to use a program called "GitHub", which can be used to synchronise our code through the internet to ensure we are always working on an up to date code. This would also make it much easier to combine code at the end as we would already be sharing variables and such. At the second meeting we grouped the functions appropriatley and then decided who would get what functions. A group member was keen to get started and quickly wrote up all the base code that would be commonly shared between us.

I was put in charge of coding the Package sorter, designing the form SorterSession and the 3 functions assiciated with the sorter. I started by drafting on paper what each function should look like, with appropriate buttons and list boxes etc. I then combined all these functions onto one form and looked at which form object could be combined. I used NetBeans to draw the form up and generate the code. Once the form code was generated, I copied it into Sublime Text 2, where I would code all the functions into. I used my form along with the base code to start coding my functions. I looked through the code and realised there were already data types and variables that I would be using, there were even a few functions another group member had written that I could use.

As I roughly designed the code (before actually coding), I encountered a few complications, including the process of scanning Packages/Bins. I decided to just let the user display the packages/bins in list boxes and make their selection (instead of scanning them), I had to go back to NetBeans and change the form in order to match.

I am very happy with how this group operated, We usually had 1-2 group meetings per week, which kept me motivated to work. Nailing down the shared code earlier was a very convieniant thing to get done. And when there were any part of that I didn't understand, a group member would quickly point out how it worked.