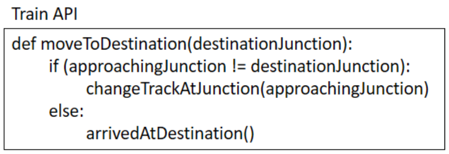
Tracy Yee Enying, 1002379

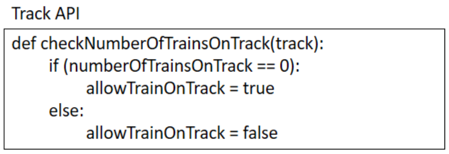
50.003 Problem Set 1

1. Cohort Exercise 1.2 – Railway Network APIs 1



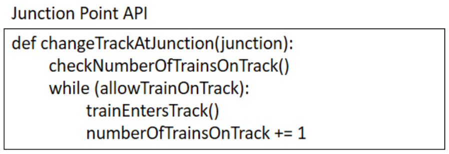
Train API

```
def moveToDestination(destinationJunction):
    if (approachingJunction != destinationJunction):
        changeTrackAtJunction(approachingJunction)
    else:
        arrivedAtDestination()
```



Track API

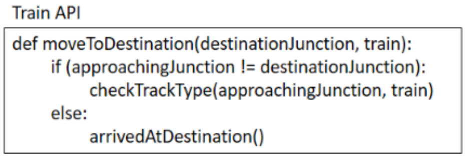
```
def checkNumberOfTrainsOnTrack(track):
    if (numberOfTrainsOnTrack == 0):
        allowTrainOnTrack = true
    else:
        allowTrainOnTrack = false
```



Junction Point API

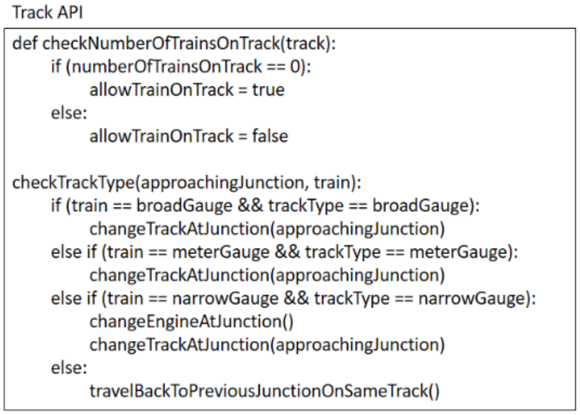
```
def changeTrackAtJunction(junction):
    checkNumberOfTrainsOnTrack()
    while (allowTrainOnTrack):
        trainEntersTrack()
        numberOfTrainsOnTrack += 1
```

2. Cohort Exercise 1.3 – Railway Network 2 APIs 2



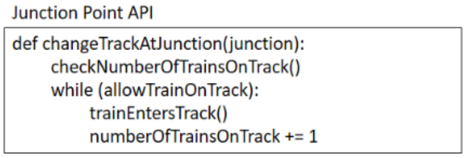
Train API

```
def moveToDestination(destinationJunction, train):
    if (approachingJunction != destinationJunction):
        checkTrackType(approachingJunction, train)
    else:
        arrivedAtDestination()
```



Track API

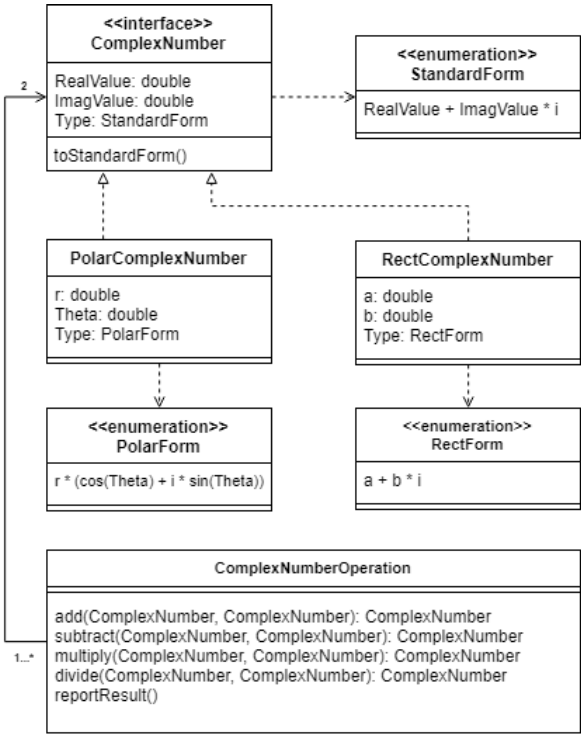
```
def checkNumberOfTrainsOnTrack(track):
    if (numberOfTrainsOnTrack == 0):
        allowTrainOnTrack = true
    else:
        allowTrainOnTrack = false

checkTrackType(approachingJunction, train):
    if (train == broadGauge && trackType == broadGauge):
        changeTrackAtJunction(approachingJunction)
    else if (train == meterGauge && trackType == meterGauge):
        changeTrackAtJunction(approachingJunction)
    else if (train == narrowGauge && trackType == narrowGauge):
        changeEngineAtJunction()
        changeTrackAtJunction(approachingJunction)
    else:
        travelBackToPreviousJunctionOnSameTrack()
```



Junction Point API

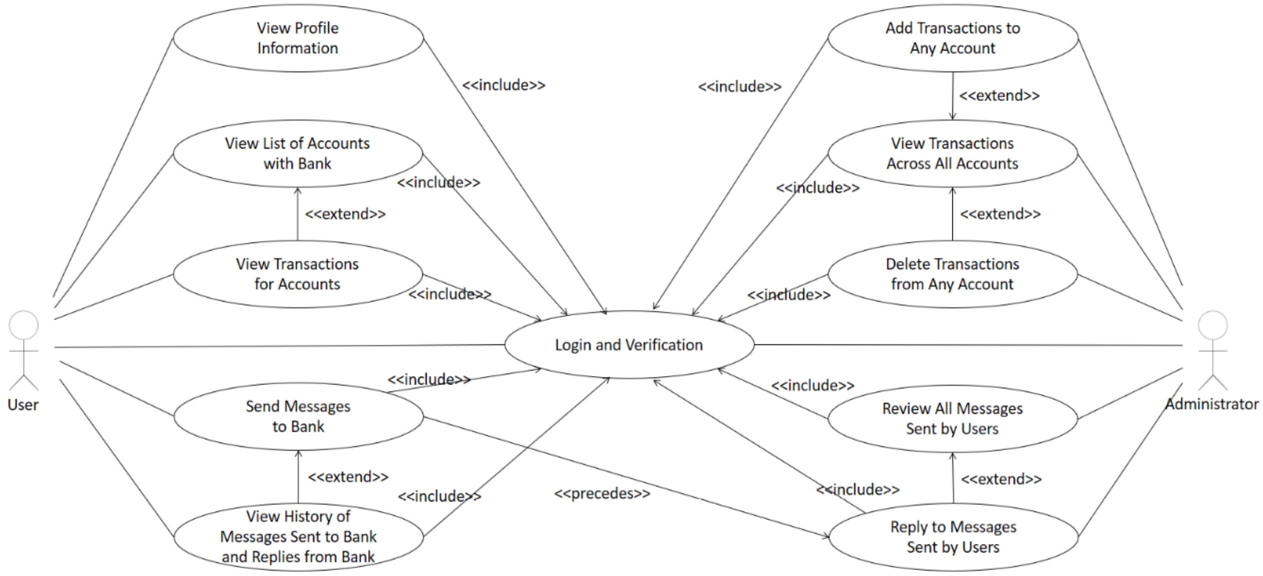
```
def changeTrackAtJunction(junction):
    checkNumberOfTrainsOnTrack()
    while (allowTrainOnTrack):
        trainEntersTrack()
        numberOfTrainsOnTrack += 1
```

3. Cohort Exercise 2 – Complex Numbers

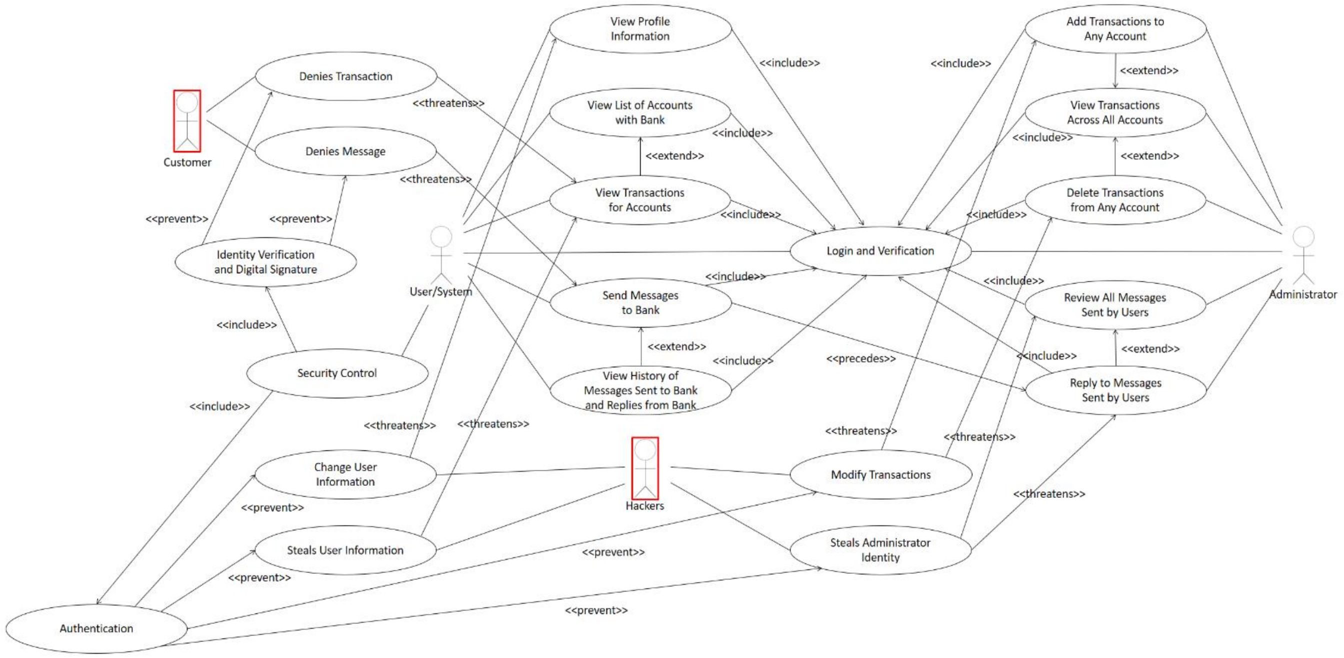


```
public class ComplexNumber {
    private double realValue;
    private double imagValue;
    private static ComplexNumber z = new ComplexNumber();

    public static void main(String[] args) {
        ComplexNumber c1 = new ComplexNumber(7.2, 3.0);
        ComplexNumber c2 = new ComplexNumber(12.1, 6.7);
        System.out.println(add(c1, c2));
        System.out.println(subtract(c1, c2));
        System.out.println(multiply(c1, c2));
        System.out.println(divide(c1, c2));
    }

    public ComplexNumber(double realValue, double imagValue) {
        this.realValue = realValue;
        this.imagValue = imagValue;
    }

    public ComplexNumber() {
        realValue = 0;
        imagValue = 0;
    }

    public static ComplexNumber add(ComplexNumber c1, ComplexNumber c2) {
        z.realValue = c1.realValue + c2.realValue;
        z.imagValue = c1.imagValue + c2.imagValue;
        return z;
    }

    public static ComplexNumber subtract(ComplexNumber c1, ComplexNumber c2) {
        z.realValue = c1.realValue – c2.realValue;
        z.imagValue = c1.imagValue – c2.imagValue;
        return z;
    }

    public static ComplexNumber multiply(ComplexNumber c1, ComplexNumber c2) {
        z.realValue = (c1.realValue * c2.realValue) - (c1.imagValue * c2.imagValue);
        z.imagValue = (c1.realValue * c2.imagValue) + (c1.imag * c2.real);
        return z;
    }

    public static ComplexNumber divide(ComplexNumber c1, ComplexNumber c2){
        double r = Math.pow(mod(c2), 2);
        z.realValue = ((c1.realValue * c2.realValue) + (c1.imagValue * c2.imagValue)) / r;
        z.imagValue = ((c1.imagValue * c2.realValue) - (c1.realValue * c2.imagValue)) / r;
        return z;
    }

    public static StandardForm toStandardForm(PolarForm) {
        z.realValue = PolarForm.r * Math.cos(PolarForm.theta);
        z.imagValue = PolarForm.r * Math.sin(PolarForm.theta);
        return z;
    }

    public static StandardForm toStandardForm(RectForm) {
        z.realValue = RectForm.a;
        z.imagValue = RectForm.b;
        return z;
    }

}
```

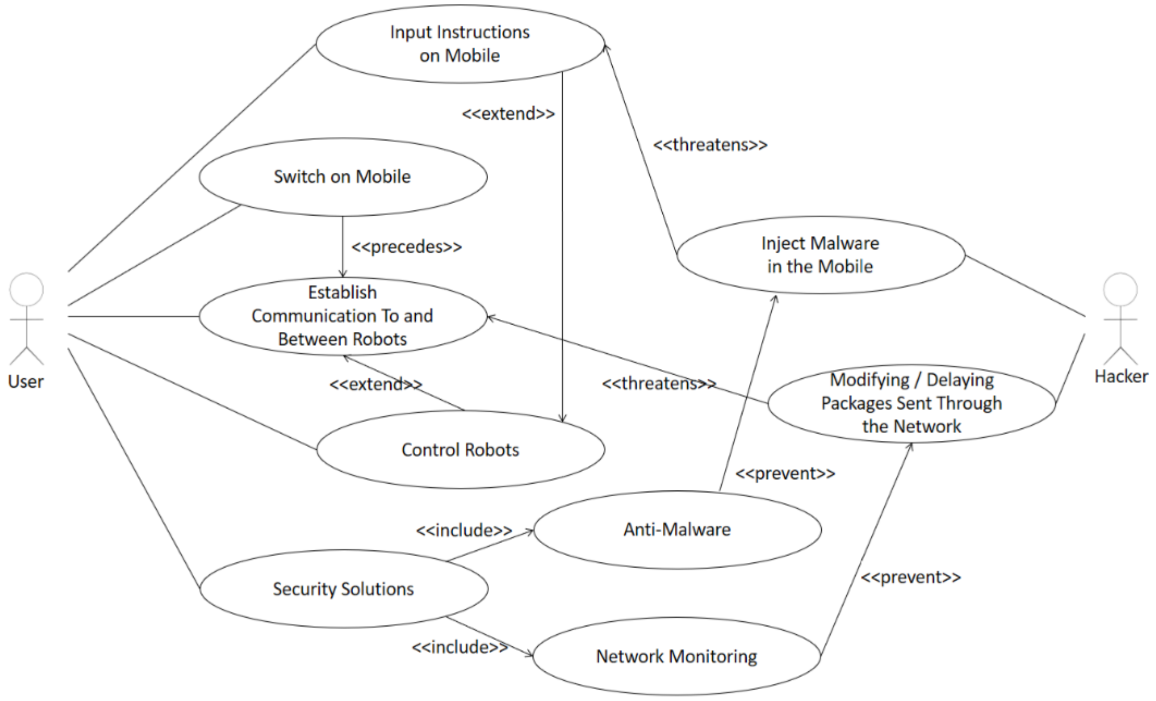
4. Cohort Exercise 4 – KBO Use Case Diagram


View Profile Information
<<include>>
View List of Accounts with Bank
<<include>>
<<extend>>
View Transactions for Accounts
<<include>>
User
Login and Verification
<<include>>
Send Messages to Bank
<<extend>>
<<include>>
View History of Messages Sent to Bank and Replies from Bank
<<precedes>>
Add Transactions to Any Account
<<include>>
<<extend>>
View Transactions Across All Accounts
<<include>>
<<extend>>
Delete Transactions from Any Account
<<include>>
Administrator
<<include>>
Review All Messages Sent by Users
<<extend>>
<<include>>
Reply to Messages Sent by Users

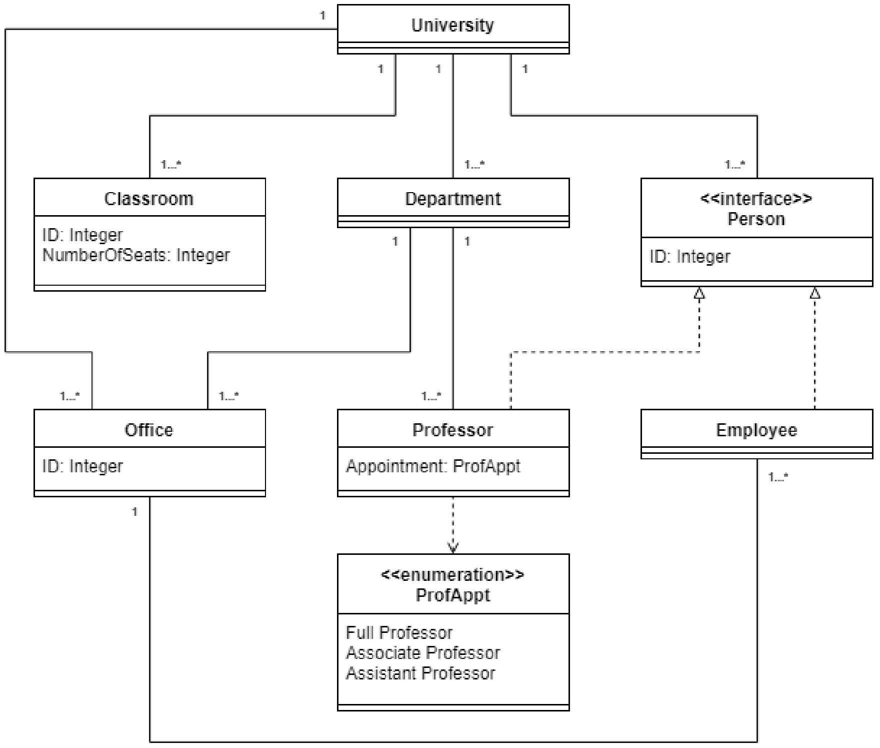
5. Cohort Exercise 5 – KBO Misuse Case Diagram



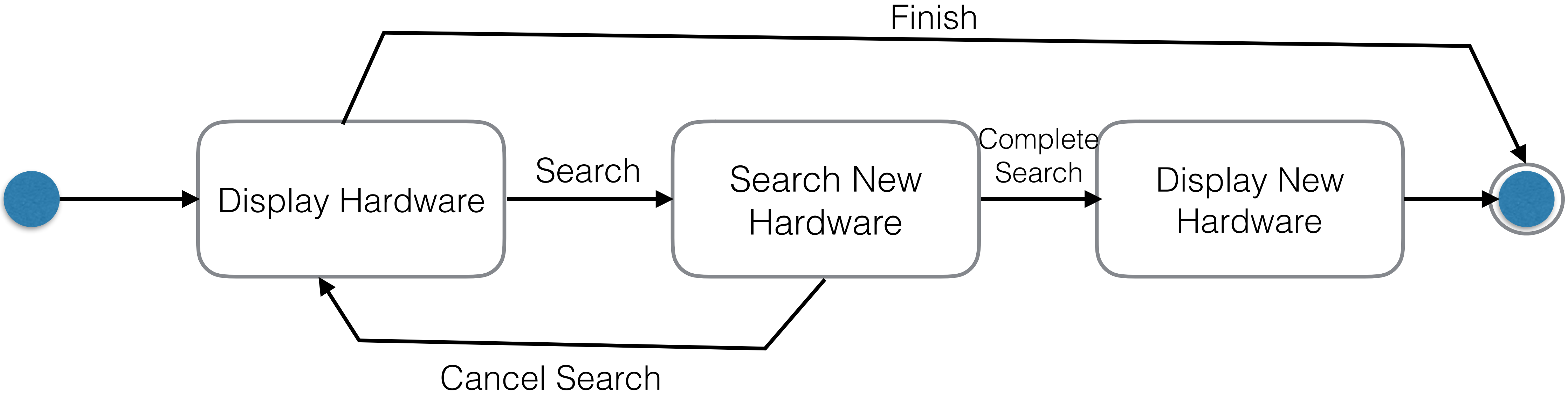
6. Cohort Exercise 6 – Robot Swarm Use-Misuse Case Diagram



7. Cohort Exercise 7 – University Class Diagram



Exercise 8


Finish
Display Hardware
Search
Search New Hardware
Complete Search
Display New Hardware
Cancel Search