

- The system will allow property owners to register each of their properties
- Owner pay the property tax due for the properties
- View a list of their properties
- View the tax that is due currently per property
- View any overdue tax (hasn't been paid for a previous year)
- Query specific previous years and get a balancing statement for any particular year or property

- The property owner should be able to maintain a record of all payments of the property charge on a yearly basis
 - history?
- A property owner will be able to view payments made for all their owned properties.
 - Current year??

- The property must be registered on the system before the property tax can be paid
- A flat charge of €100 if the property is not the principle private residence of the owner.
- Apply a 7% penalty, compounded for each year that a property tax is unpaid.

Initial Design

Initial design discussed equally as a group

Noun/Classes(possible)

Property (Tax variable)

Calculation class

Verb/Methods

Owner can View a list of their properties

Department can view list of properties under certain headings

Register property

Pay tax


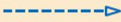
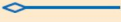
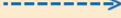
Get balance

Access previous records (csv file being read??)

Register owner

Find tax overdue and apply penalty

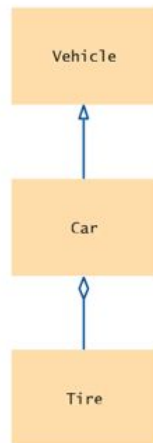
Be able to get statistics (only needs to be available to department) this is based off eircode

Relationship	Symbol	Line Style	Arrow Tip
Inheritance		Solid	Triangle
Interface Implementation		Dotted	Triangle
Aggregation		Solid	Diamond
Dependency		Dotted	Open

Example

```
class Car extends Vehicle
{
    ...
    private Tire[] tires;
}
```

Figure 6
UML Notation for
Inheritance and Aggregation



Data fields for property class

Property owner(s)

Address

Postcode/Eircode

estimated market value

location category (City/Large town/Small town/Village/Countryside)

Principal private residence (yes/no)

Method: append new property into the property csv files

Need exception handling for reading csv files

Probably need exception handling for user input = Built in handling in Scanner class

Payment arraylist (payments made by owner in the array, if not there, the payment is overdue)

For eircode

Eircode is 7 characters long

First 3 is the routing key to identify the area

Possibly have two classes with main method

One of the classes has main method for running from command line and taking arguments from command line

The other class with main method implements the GUI

Responsibility of class:

Property class

Read csv file

Property History

Getting previous info on property

Show that there is overdues

Department Personnel class

Do statistics in here

Tax Calculator class

Impose penalty for overdue tax

Payment class

Records amount

The year paid

Owner class

Pay tax and add to list of type payment

