

## CS4013 project breakdown

Objective:

To Develop Property Charge Management System

### System Operation:

1. Allows owners to register each of their properties & pay tax on it
2. Its due to be paid Jan 1<sup>st</sup> each year
3. Owners should be able to view a list of all their properties and tax that is due / property & an overdue that has not been paid for previous year
4. Owners should also be able to pick specific previous years and get a balancing statement for any year on that property.
5. System should keep record of all payments of the property charge on yearly basis
6. All records should be held in a CVS files
7. A property owner will be able to query the system to view payments made for all their owned properties.

**To Register:** A property must be registered on the system before the tax can be paid. Owners can register their register their property by providing the following.

- Property owner(s)
- Address
- Postcode/Eircode
- estimated market value

- location category (City/Large town/Small town/Village/Countryside)
- Principal private residence (yes/no)

**Calculate Property Tax:** its calculated based on the following.

- Fixed cost of 100 euros
- Market value tax based on the following rates:

Property Value	Rate
Up to 150,000	0
150,000 - 400,000	.01%
400,001 - 650,000	.02%
Above 650,000	.04%

- Location category based on the following rates:

Location	Charge
City	€100
Large town	€80
Small town	€60
Village	€50
Countryside	€25

- Plus 100 euros flat charge if the property is not the principle private residence of the owner
- Plus, a 7% penalty, compounded for each year that a property tax is unpaid

## **Management Functionality for the Department of Environment:**

the management should be able to do the following:

- Get property tax payment data for any property
- Get property tax payment data for any property owner
- Get list of overdue property tax for a selected year/ with option to select an area based on either the routing key or Eircode.
- Get property tax statistics for a particular area based of Eircode or routing key. E.g. total tax paid, average tax paid, number and % of tax paid.
- Investigate the impact of possible changes to the rates and levies contributing to the property tax to determine how the revenue collected would change.

## **Other Requirements:**

- Command line Interface and GUI should be provided. The system should be designed that its easy to replace one interface with another.
- The developed system should be adaptable to other property tax calculation rules (e.g. Other countries) therefore want the design so that it is easy to substitute the tax calculator with a different one in the future.

## Check list and deliverables:

1. A document outlining the Class Responsibility Collaboration (CRC) cards and a UML diagram showing the relationships between the classes.
2. Documentation for the software generated with Javadoc utility and a helpfile describing how to compile and run the application from the Command Line interface. Docs should also include link the GitHub.
3. Source code for the system // each java class is stored in a separate file //All code will be run and compiled from command line – must make sure that instructions for execution are clear and accurate.
4. Any text files required by the system must be included e.g. files storing appropriate test data // this should include data for at least 10 properties with at least 3 owners and 3 different postcodes or routing areas. These files should be stored in an appropriate subfolder(s) to facilitate direct program compilation and execution when stored on a different machine // do not include any other files e.g. configuration files for a particular development environment.
5. Contributions document: where contributions of each team member must be clearly outlined and agreed on by all team members. Each member overall contribution// documentation// implementation// and testing must be stated. In addition, the author of each class and method must be specified along with the authors of the design and documentation.

## Work Breakdown:

1. Develop *main* class and subclasses that may be required
2. *Calculate tax* class & *to register* class and subclasses that may be required.
3. *Management functionality* class and subclasses that may be required.

4. *Other requirements* e.g. interface ... and CRC card & UML diagrams