**CS4013 Project Contributions Page**

Members: Jack Busher, Dean Kelly, Leon Woods, Eoin Chedzey, David Barrett

CommandLineUI.java – Dean Kelly

Department of Enviroment Record – Leon Woods

Gui.java – Dean Kelly, Jack Busher, Eoin Chedzey

Property.java – Jack Busher, Dean Kelly

PropertyManagementImpl.java – Dean Kelly, Eoin Chedzey

PropertyManagementInterface.java – Dean Kelly

Tax.java – Jack Busher, Dean Kelly, Eoin Chedzey

Test.java – Dean Kelly, Jack Busher

propertiesTax.csv – Eoin Chedzey

CRC Cards – Leon Woods, David Barrett

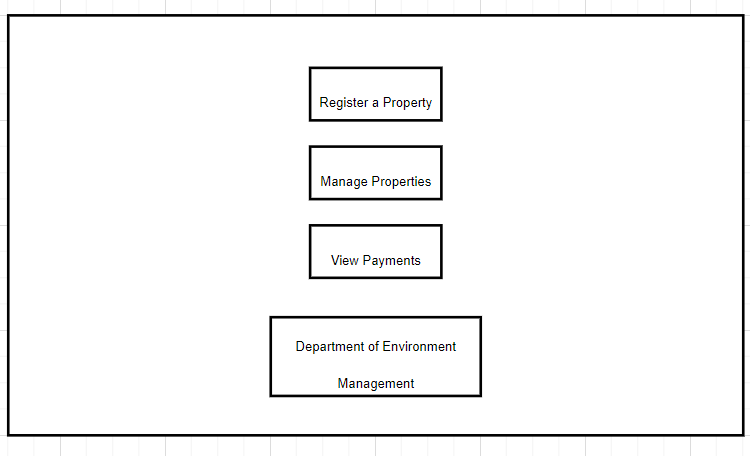
UML Diagram – David Barrett

**Eoin Chedzey**

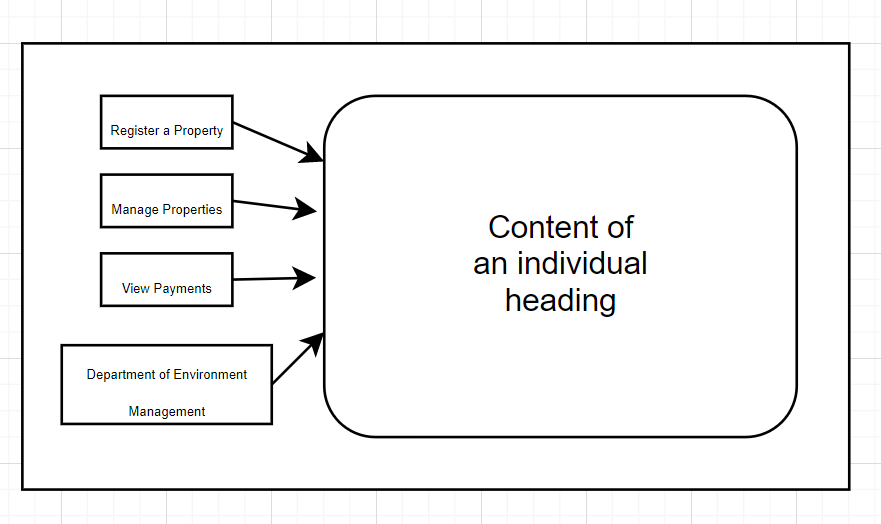
**ID:18223796**

**Design**

I focused a lot on the Graphical User Interface design. I initially started sketching up layouts for the GUI, starting off with a home screen with 4 individual JButtons. Each button would then lead the user to a different page in order to carry out the task that they had chosen.

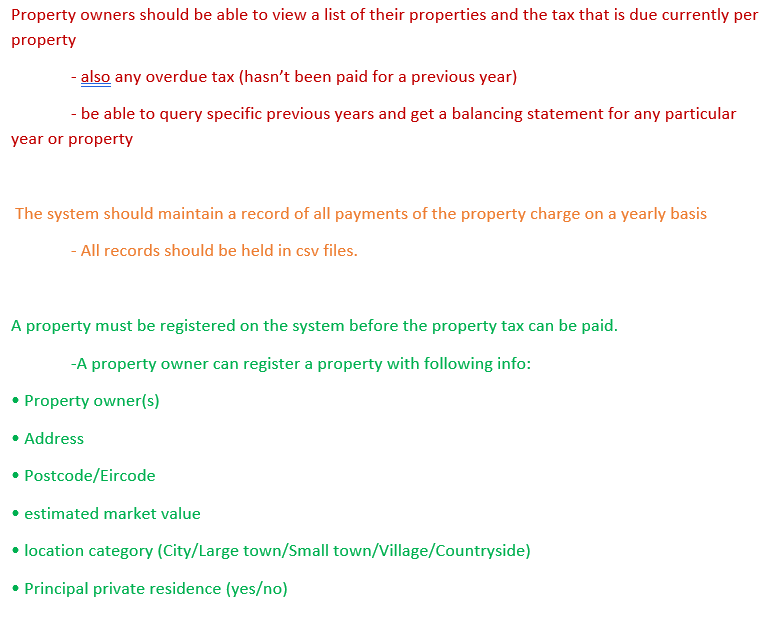


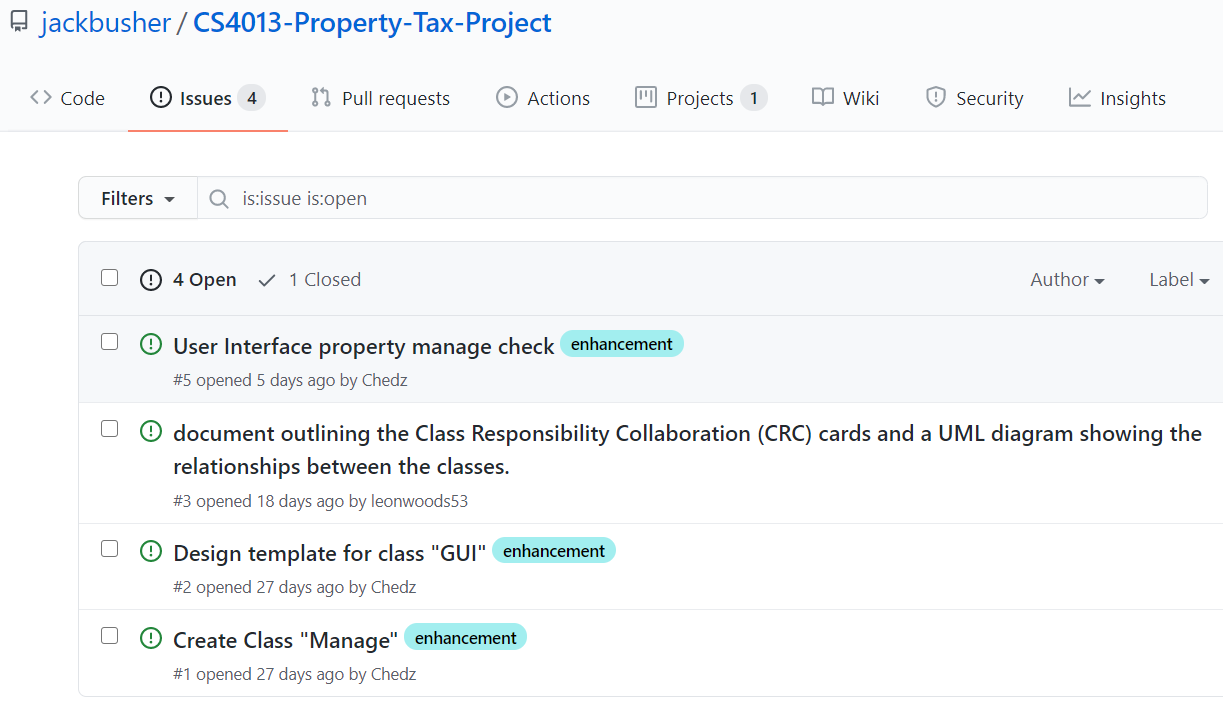
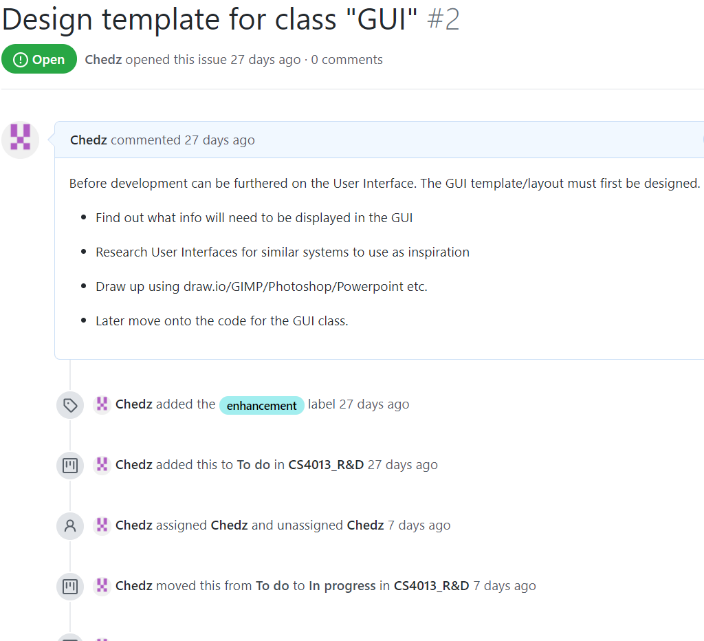
However, I decided to take a slightly different approach. I split the GUI into two different Jpanel’s. The first panel would be on the LHS of the screen with the 4 Header Jbuttons. The panel on the RHS would change depending on the last Jbutton pressed.

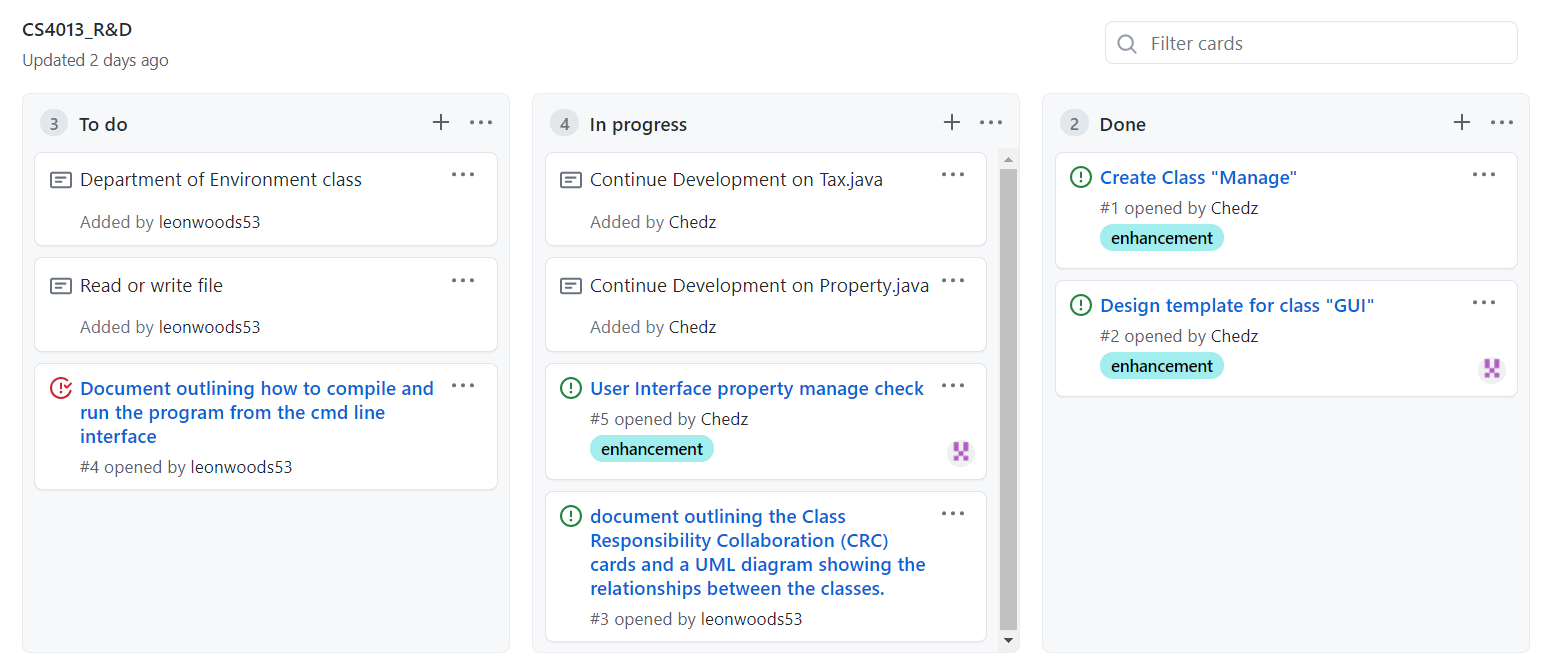


**Documentation**

I first broke down the Project specification into simplified point and inserted them into a word document. I tried to change the colour of the text to reflect what had and had not been done (green=done, yellow=in progress, red=not started). This did work but made it difficult to add in new issues and to share issues with the rest of the group.

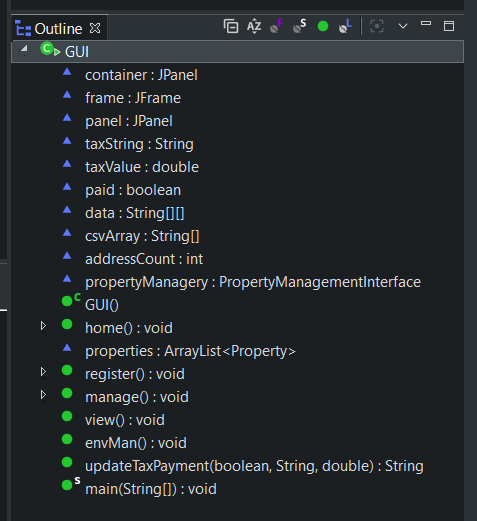


I started to document certain issues on GitHub. I gave a brief description of what needed to be done and what it hopes to achieve.

This led to me implementing a Kanban board to manage these issues and assign to each other.

The Kanban board with documented issues was definitely an improvement on the colour coded word document. Using these tools along with the repository itself, this was the first software development project that I have relied on one whole system to store, update, and manage everything.

**Implementation**

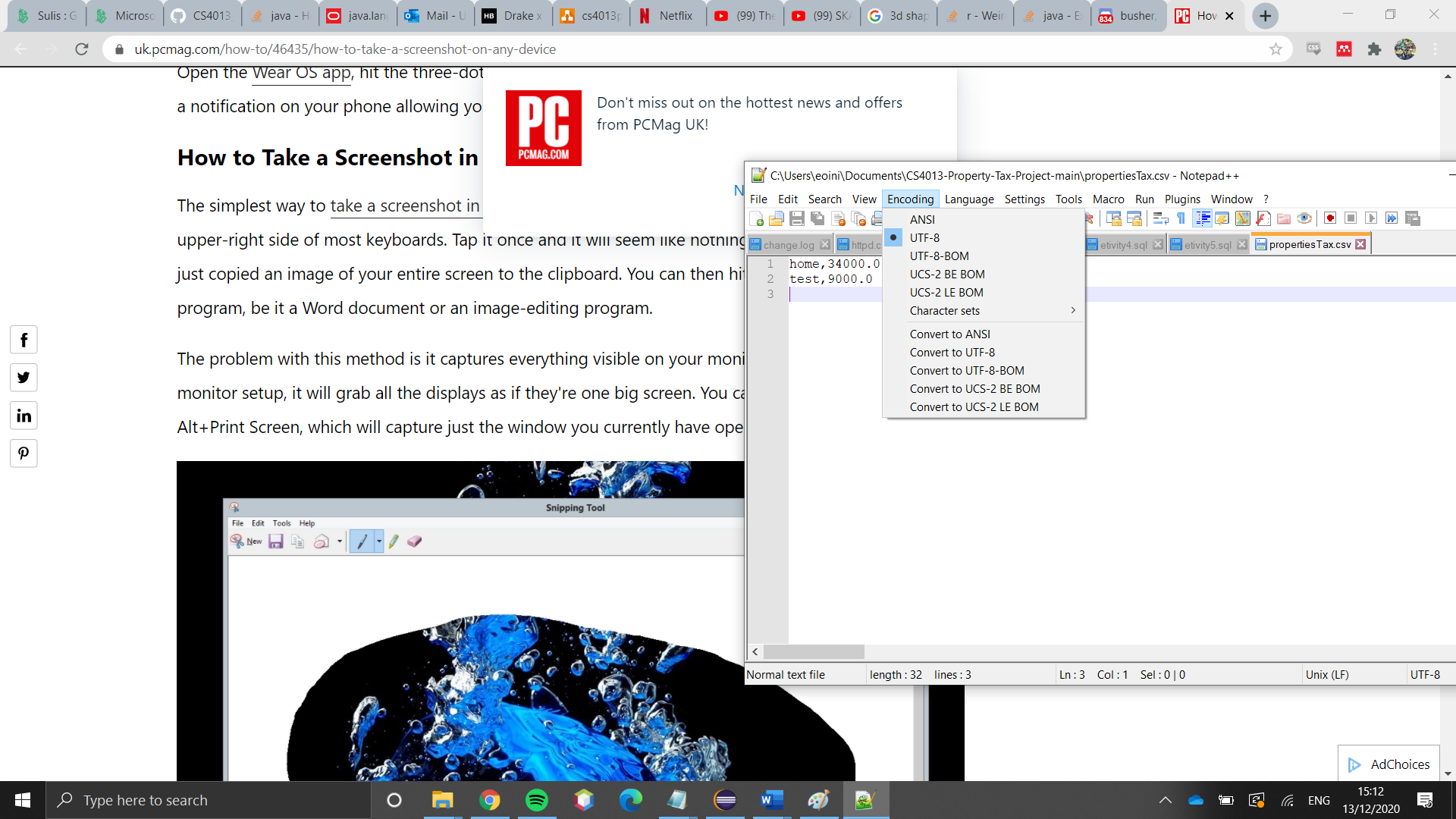
Coded most of GUI.java and also coded part of Tax.java and Property.java

I focused mainly on the implementation of the User Interface. I was the primary contributor to GUI.java, which was built up by myself, Jack and Dean. I also contributed a respectable amount to Tax.java and Property.java in order for it to interact with the User Interface appropriately.

I devised and programmed the methods: home(), register(), manage(), view() and updateTaxPayment(). I am not entirely happy with the updateTaxPayment() method. It is designed to read and write all property tax payments and outstanding balance to the file: propertiesTax.csv. However, it ended up becoming over complicated and still has some issues with updating the csv file and content ‘drifting’ into other cells. I am sure a better approach could have been taken, for future reference I would have split this method up into smaller methods all contained in one class exclusively for managing propertiesTax.csv.

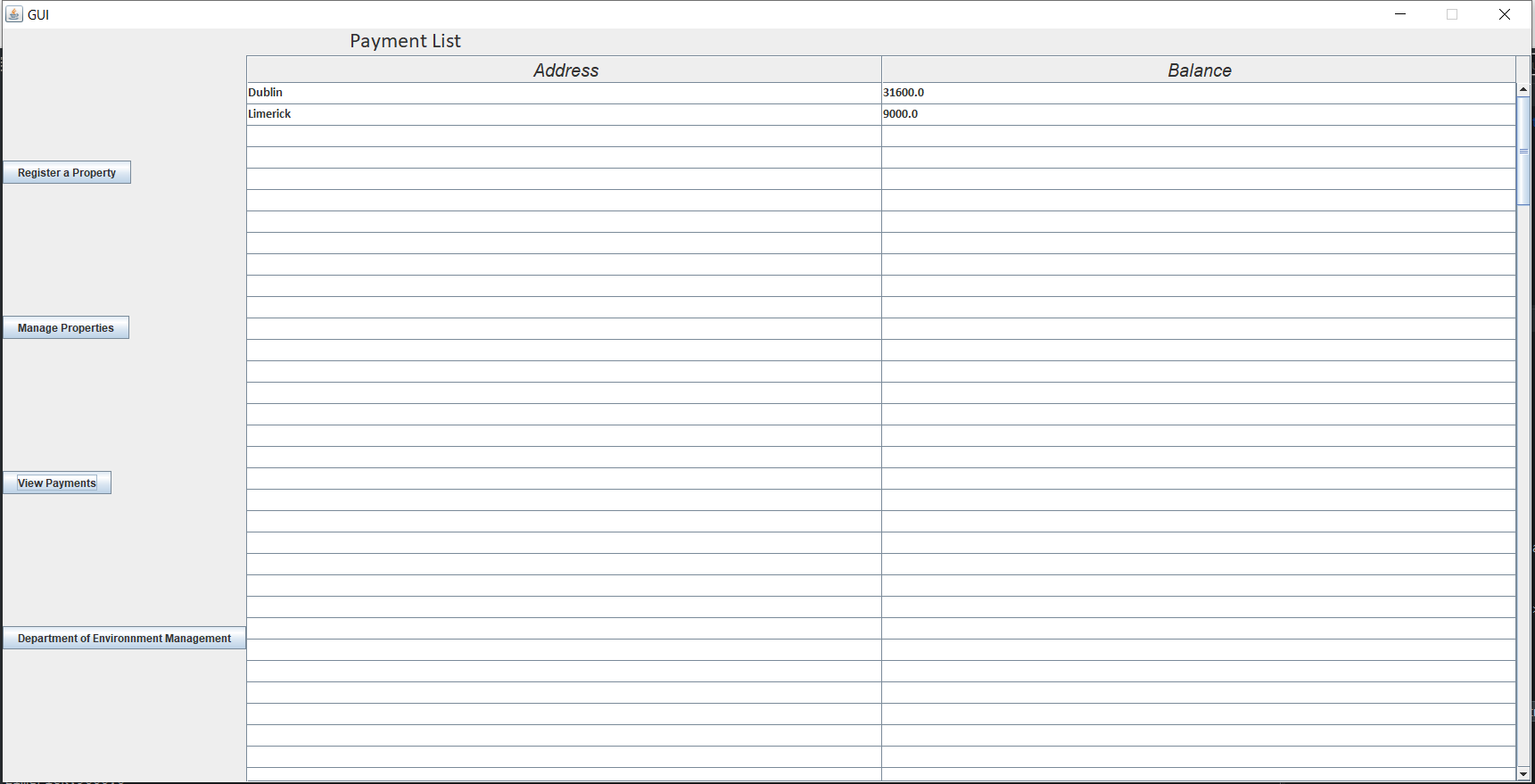
**Testing**

I manually tested the User Interface a lot while implementing GUI.java. I understood how these methods read and wrote to the csv files so I would try to purposely ‘trick’ the system by adding new properties, calculating their tax, creating tax payments for multiple properties which would in turn update and overwrite the propertiesTax.csv file. This resulted in a multitude of errors from IndexArrayOutOfBounds exceptions to overwriting incorrect data and not writing data to the csv file at all.

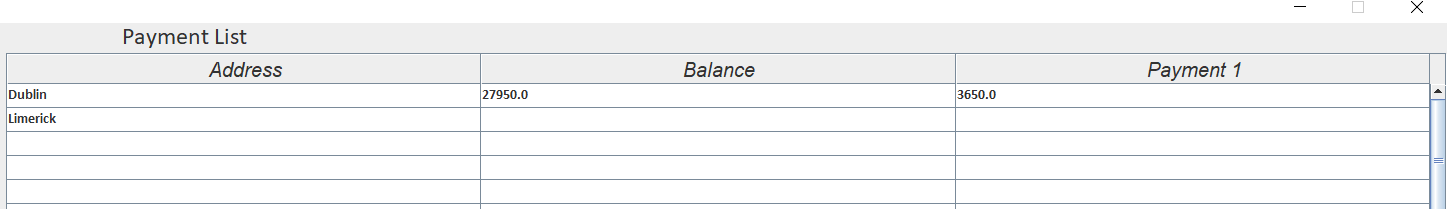
This was my first time dealing with csv files and using comma delimited cells has been a frustrating but rewarding experience. I even learned that different types of encoding affect the content of the csv file. I had to change the encoding manually from the one created and saved in excel to UTF-8. Originally it contained BOM characters “**(ï»¿)**” in the first cell, I later realised that excel supports this output format.

There is still one issue that still happens when testing and could not be resolved.

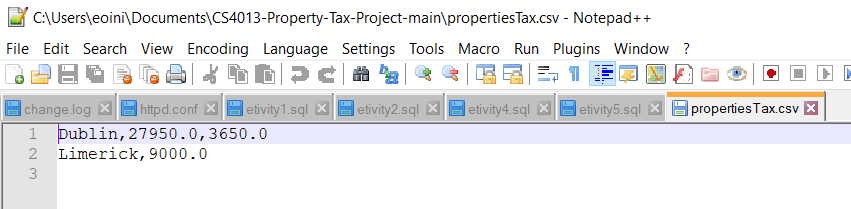
After registering 2 properties and calculating tax for both of them, they are added to the Jtable perfectly.



When a payment is added to either property, the final property has its tax balance disappear from the Jtable.



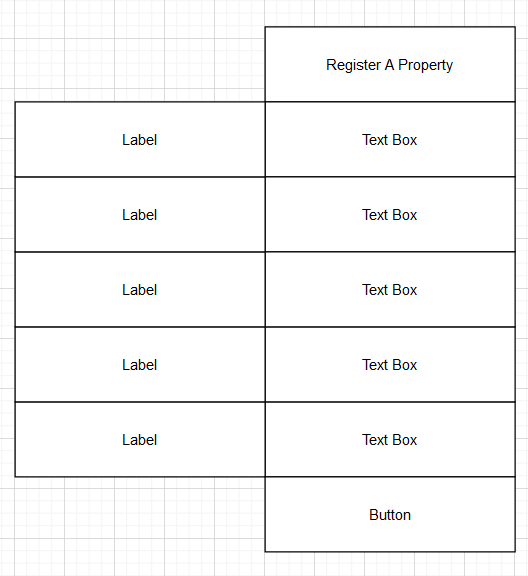
Interestingly, the double is still present in propertiesTax.csv when opened in Notepad++ or Excel.



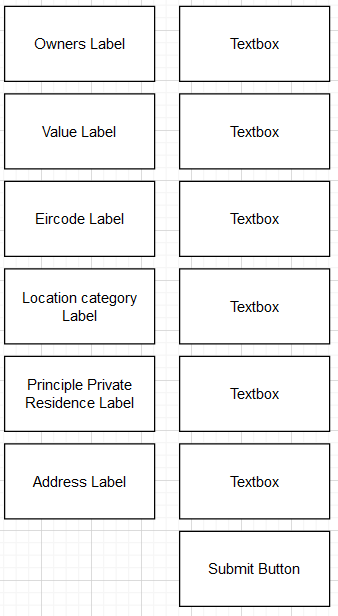
It’s value is read by the reader into the array that is then supposed to be writing back into the Jtable. However, its value is not writing back in. I believe this could be solved by splitting the updateTaxPayment() method into smaller methods in one new class as mentioned earlier, time constraints resulted in an unpolished product but not through my lack of effort.

Jack Busher 19237499 – Contributions to Design, Implementation, Documentation, Testing

**Design**

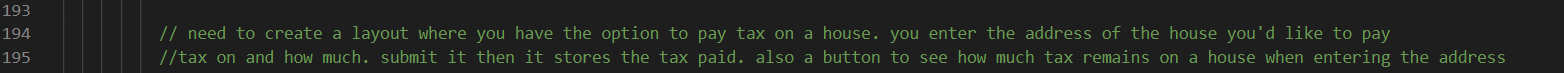
I initially designed the ‘Register a property’ on draw.io. I wanted to make it a simple and clean GUI to use. This was my first draft. 

I also designed the ‘Manage Properties’ section. I wanted it to look similar to the ‘Register Property’ section but I needed to change it a bit because there was going to be more than one function on this page. This was what I came up with at the beginning.



**Documentation**

I added comments in the code that I developed. This was to help the other members in my group understand easily what I had added. I also added comments where code needed to be implemented.



**Implementation**

I developed a good amount of code. I developed a lot of the GUI. I mostly created all the Swing objects and implemented them so that it looked good. The GUI was a combination of Dean, Eoin and my work. I completed the Property class by myself, and developed a lot of the Tax class. Before the project I didn’t have much experience with developing GUI’s. During the course of this project I learned a lot about GUI’s.

**Testing**

I was constantly testing the updated code whenever someone uploaded anything. One thing I wish I had done was upload the results of the tests and maybe some sort of evaluation of the tests. This is something I would do in future if I was to do this project again.

**Dean Kelly.**

**19272898**

**Design…**

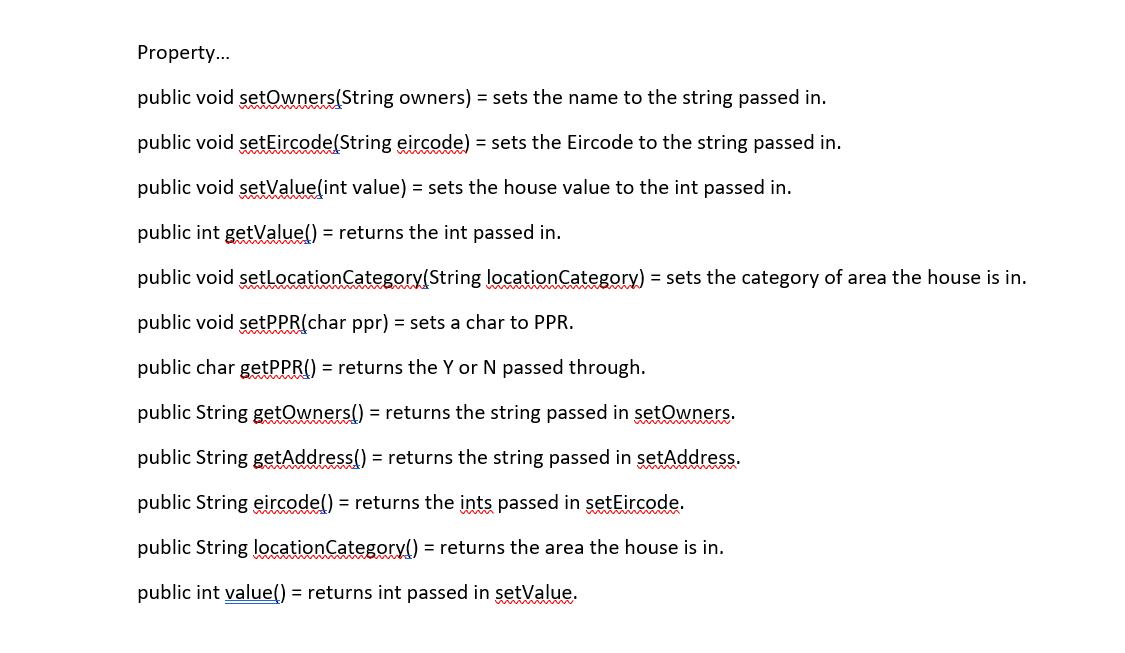
I had initially created a GUI but we went with Eoins and decided as a group his was better suited. Table

Description automatically generated

This GUI is User friendly and easy to navigate which is what we wanted.

**Documentation…**

Most of the documentation I created was to keep track of whats done and what needs to be done. I also created a list of what each method does in the Property and Tax classes.



**Implementation…**

Property Management Interface….

I created this interface to allow these abstract methods to be implemented. This is linked with Gui and is how we connect to it.

GUI…

Jack, Eoin and myself were working on the GUI together. If I could go back on the project I would have done this differently as our GUI turned out to be somewhat overwhelming because all 3 of us were adding to it.

PropertyManagementImpl…

I created a getProperties() method in this class. Every Property that was passed through went to a csv file called “properties.csv” returns

I then created both getTaxForOwner and getPropertiesForOwner methods. One method returns the tax on a property and the address, while the other returns the property tax with the owners name.

Initially we had another class for these method but I decided to move them into properties. I couldn’t get this to completely work. When department of environment is called the tax owed is put in a list but is not formatted.

Tax…

This class was worked on by Eoin, Jack and I. I regret how we made this class. We made it extend the property class when it didn’t need to. We decided not to change it as the project had progressed to far on by the time we realised.

**Testing**

When a method is created or a change was made to the GUI it was tested straight away. Some methods didn’t work initially even though they compiled. Through minor changes usually they were fixed and tested. Overall project testing was a big part.

**David Barrett 19264933**

* I was responsible for CRC cards. I also did a UML diagram It helped with easier and clear understanding of the overview and the schematics involved with the overall project

**Leon Woods (19251726)**

* I was responsible for the Department of Environment Class, which provides management functionality. I also created the CRC cards and I helped David with the UML diagrams.