Report for Group 26 CS1013

Members of the Group:

Eoin Dowling, Niall Hunt, David Kilroy, Chihun Lee

Introduction:

The goal of the project was to construct an application to explore data on property transactions in processing. The data set we used is from the UK Land Registry.

The application we designed had an intuitive visual design which allowed the user to fetch available queries of data, which will be then displayed to the user in a aesthetically pleasing and most importantly informative manner.

For the project the csv file which contained all the data was processed and manipulated and put in a database where SQL queries could be used to fetch data quickly and efficiently.

Member Roles and Team Structure:

Each member had a loose role and job title. We mainly stuck to these roles however, the majority of the problems were solved as a group.

Eoin Dowling: was our team leader. Eoin worked on the UI and refaced our program. The program is based on his designs. He also worked on updating and integrating the screen class. He added fonts, images, sound files, and created a colour scheme for the program. Additionally, Eoin created a working Search Bar with the help of a library called Interfascia. This library turned out to be very awkward to use as creating a search bar that would disappear when clicked out of screen was a challenge. In the end, Eoin found a solution by creating and deleting the GUI controller dynamically.
Eoin helped keep us organised as he created a group chat and a wunderList.

Niall Hunt: Niall created the BarChart, PieChart and Histogram classes. He also provided the templates and basis for the widget and screen classes. He was mainly in charge of the data visualisation but helped with the queries, screens and widgets. He worked closely with David to help the queries to display the correct data.

David Kilroy: was tasked with creating the Queries and Statistical measures for the program. He implemented the drop down menu during the first week, and worked with Niall on the widget class. David was universally resourceful and helped make sure everyone's contributions integrated probably with each other. He worked extensively on the query class and screen system. His work pulled all of the pieces together and allowed the program to run.

Chihun Lee: was charged with “back-end development”. Chihun learned how SQL databases worked and implemented a database into our project. This allowed for our programs data to be organised easily and added speed to our program. He provided assistance to everyone on the “front-end” coding too, such as the Queries and search bar which required integration with the program.

Chihun also implemented the map of the UK with the use of the library geocentre.

Code Structure:

Screens:

The layout of the program is based around a number of different screens which navigate to the desired search location and to view a set of queries related to that query. The user can search by specific town, by county (using the map) and by district. Each of these buttons brings the user to the query screen relating to the chosen location. The query screens display the data of a location using appropriate bars and graphs.

Widgets:

The widget class was written by ourselves. We used what we had done in lab 6 as a template for this class. It is given an x and y position and a width and height. A method within this class will check the mouseX and mouseY every time it is called. If the mouseX and mouseY are over the widget it will return a given int value as an event. This allowed us to create buttons, drop down menus and allowed the bars of the bar chart to display data when hovered over.

Queries:

The queries work on the basis that a user would first choose what data set which they would like to use(county, district, town, all of uk) and then the users could choose different queries on the data set and these queries would be displayed onto the Screen.
The queries were handled by the query class. The query class would take in the search and optionally the type of data the user wanted to use and the query class had different methods which would fetch the queries and display them on screen.

Features:

Toolbar:

Back Button:

The back button utilises an ArrayList which contains all of the previous screens the user has viewed. When the back button is clicked the previous screen is removed from the list and the current screen changes to the previous screen. Measures were taken to ensure that when the back button was pressed nothing was drawn that wasn’t intended to be displayed on the page from the previous page.

Home Button:

When the Home Button is clicked the current screen is changed to the default Home Screen which brings the user back to the start of the program. As with the back button, measures were taken to ensure that when the back button was pressed nothing was drawn that wasn't intended to be displayed on the page from the previous page.

Drop Down Menu Bar:

When the mouse hovers over the drop down button a menu appears under the button which contains a set of widgets. Originally intended to be used as the primary interface for selecting Screens and Queries in our first, we late decided to use these the drop down menu to navigate to the 'About Us' screen, as well a link to the our repository and a button to play an appropriate tune.

Visual Data Representation:

Barchart:

The barchart class was created to represent a given float array visually. It has many different constructors to allow the others in the group to use it easily and to suit their specific needs. It takes in an x and y position and width and height values. This allows it to dynamically change size to fit our screen. The size of the bars are all calculated based on the inputted data. A label can be added and the x values can be displayed when hovered over

Pie Chart:

The PieChart class much like the BarChart class and the Histogram class allows others to create a pie chart with different sizes and x and y positions. It calculates the size of the segments based on the inputted data. A key can also be added which shows the users a colored square that corresponds to the data in the chart.

Interactive Map:

The interactive map allows to select a county in the United Kingdom or where the user can then see data of the chosen county. We intentionally disabled the selection options for Scotland and Northern Ireland as the land registry data only included data. The user can select a county by pressing the cursor on the county on the map. The county will also highlight if the user hovers over the county and will display the name of the county beside the county on the map.
The interactive map was implemented with the help of the library 'Gicenter.utils', particularly the Geo Map library as well as shapefiles and .dbf files which are used to found the data on the map and its counties shape and other details. The shapefiles were sourced from(www.gadm.org).

Search Bar:

Other Features:

About us Page:

The ‘About Us’ page is a screen that gives info to the users about us(the developers of the applications). It is accessible from the drop down menu. It was created by Eoin Dowling using the Adobe Suite after he determined it to be the simplest solution for creating a page that contained all the contact information, such as email address and social media links.

Link to Our Repository:

A web-link to our repository is available from the drop down menu. It make use of the link() function in processing and will open using the computer’s default web browser program.

Option to Play ‘God Save the Queen’:

You can play the UK national anthem ‘God Save the Queen’ from the drop down menu. It was implemented using the library Minim. The purpose of this was to demonstrate the potential for sound and audio should it be required.

Unused Features:

A number of classes and methods were designed, created and fully functional but were not implemented in the final product. The functionality of these cut features and the reasons for their exclusion from the final product are explained below.
All the features below

Histogram class:

The histogram is a class that can takes in a piece of data and displays it onto the screen. This class was fully completed to satisfaction however we could not find a use for it (Most of the queries we planned to use with the histogram worked better on the var chart). This worked much like the BarCharts as it changed size based on the inputted data.

PropertyEntry:

The PropertyEntry class is a class whose object that has all the information on a certain property entry. This class was never used however as we failed to find a nice and useful way to display text on a screen without just a bunch of text on a screen or table.

Unused Queries:

getPropertyEntry() is method that was originally planned so that if a user entered the address of a property entry(the house number and/or name as well as street (e.g 84, Clough Close)) the method would return an object of property entry which contained all the details of that property entry. This method was never used due to the same reason the PropertyEntry class was never used.

getStreetEntries() is method that was originally planned so that if a user entered in a street in the search bar the method would an arrayList of property entries. This method was never used due to the same reason the PropertyEntry class was never used.
displayAge() displayed whether the house was newly built or an existing property. We found however that the pie charts we made from these were not very interesting as almost all pie charts would be almost completely dominated by the older houses.
Other queries such as **Min, Max, and range** were the used as we had queries that were too similar and more interesting(such as Top 10, bottom 10).