**Estate Agency (Auctioneer)**

**Project – Year 2**

A cartoon of a person holding a gavel

Description automatically generated

**Course: Software Development (CW\_KCSOF\_B)**

A black and grey logo

Description automatically generated

**Table of Contents Page**

[**To Do** 3](#_Toc189085318)

[**Summary** 5](#_Toc189085319)

[**Entities** 6](#_Toc189085320)

[**Entity Relationships** 7](#_Toc189085321)

[**Level 1** 7](#_Toc189085322)

[**Level 2** 7](#_Toc189085323)

[**Styling** 8](#_Toc189085324)

[**Menu Style - Code** 9](#_Toc189085325)

[**HTML Structure** 9](#_Toc189085326)

[**CSS Stylesheet** 10](#_Toc189085327)

[**Key Dates** 11](#_Toc189085328)

# **To Do**

Lab: 23/01/2025

**Step 1:** - Level 1 ER Diagram (Confirm the entities on our Access prototype and draw up diagram.

* Level 2 ER Diagram (Simplify the many to many relationships & post the keys where relevant)

**Step 2:** Start designing the menu system.

**THINGS TO CONSIDER**

A screenshot of a computer

Description automatically generated**Menu Type:** Top Navbar:

Sidebar:

A white background with black dots

Description automatically generatedDiscuss the pros and cons of each menu type. Eg, the sidebar is much better for economy when considering how easily accessible each menu element is. Versus a top navbar being narrower and more condensed. Which is better for a tired, hungry staff member to use?

Do we add in extended dropdown menus when the user hovers over a menu element? Or do we keep it simple?

**List of styling options to consider**

* Colour scheme (Primary colour, font colour)

**NOTE:** Remember, the software is for the staff of an Estate Agency, we don’t need in your face style UI. Keep it simple and easy to look at.

**Also to remember, it’s an *application,* not a website**

* Font
* Button styles (Rounded edges, dropped shadow, background, colour change on hover?)
* **VARIABLE NAMING CONVENTIONS?**

Make a design document (or keep this as the primary working document) similar to how we are doing things in Systems Analysis. Remember, this is going to be our main talking point for jobs. So maybe showing that we took initiative to adhere to a Software Development Cycle of working.

Finally, remember that the ER Diagrams are due **February 3rd (Which is a bank holiday)**.

Getting the ER Diagrams done ahead of schedule means that we get allocated our screens early, which is a good idea to do, so we can know what type of work we have on our plates then for the remainder of the Semester for this module. Meaning, we don’t have that stress lingering over us

# **Summary**

This is a specification for a second year computing project based on an Estate Agent’s system. The project involves designing the interface and coding the functionality of the set of screens specified in this document. Details of the project time scale etc. are available from your tutor.

The Estate Agent’s system caters for the following:

* Bidding on property
* Buying property
* Inserting, viewing, updating and deleting client information
* Inserting, viewing, updating and deleting property details
* Producing useful reports about clients, properties and bids

The screens in this document are only used to demonstrate the required functionality. They are definitely not intended to suggest how the system should look. It is the task of the team to design a user interface that is as user-friendly as possible while still providing the required functionality.

It is the team’s responsibility to ensure that only valid data is keyed in and stored. Therefore appropriate prompts, warnings and error messages must be supplied by the system so that the user is made aware of the exact type of input that is required, its correct format etc.

On each data entry screen (any screen in which the user is asked to supply data), it must be possible for a user to change his/her mind about using the screen even after supplying data. Therefore, there must be some way of abandoning the screen without any changes being made to the stored data.

Once a user has finished entering data on a data entry screen, he/she must be asked to confirm that the details supplied are correct. If so, the relevant database tables are updated as appropriate. If not, the user is given the opportunity to edit fields on the screen.

In the case of Amend and Delete screens, a list box or a “Find” option should be used to provide fast access to the required record, rather than having to browse through a long list of records.

It is regarded as good programming practice to provide Help with all screens.

# **A screenshot of a computer Description automatically generatedEntities**

# **Entity Relationships**

## **Level 1**

## **Level 2**

# **Styling**

|  |  |  |
| --- | --- | --- |
| Element | Style notes | Code value |
| NAVBAR | Left-aligned | Display: flex; flex-direct: column; |
| Primary ColoUr | cadetblue | #5e9ea1 |
| Secondary ColoUr |  |  |
| Font | Franklin Gothic, Medium, White | font-family: 'Franklin Gothic Medium'; |
| Button Style | Cadetblue, reduced opacity  Hover – brighter shade | background-color: rgba(143, 190, 192, 0.692); |
| Heading Size | H1: 32px, H2: 24px | font-size: 32px; |
| NOTES | Exit button push to bottom, for workflow priority, ensuring user doesn’t accidentally exit application | |

# **Menu Style - Code**

## **HTML Structure**

A screen shot of a computer program

Description automatically generated

## A screen shot of a computer program Description automatically generated**CSS Stylesheet**

# **Key Dates**

A screenshot of a schedule

Description automatically generated

