2/27/2021

Ronan Behan

Higher diploma in science in Software Development

Product Design Specification

Contractor Contactor

# Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version # | Implemented By | Revision Date | Approved By | Approval Date | Reason |
| 1.0 | X19141815 | 27/02/21 |  |  | Initial Draft |

Table of Contents

[Version History 1](#_Toc528342615)

[1. Introduction 3](#_Toc528342616)

[1.1. Purpose of The Product Design Specification Document 3](#_Toc528342617)

[2. General Overview and Design Guidelines/Approach 3](#_Toc528342618)

[2.1. Assumptions/Constraints/Standards 3](#_Toc528342619)

[3. Architecture Design 4](#_Toc528342620)

[3.1. Logical View 4](#_Toc528342621)

[3.2. Hardware Architecture 6](#_Toc528342622)

[3.3. Software Architecture 6](#_Toc528342623)

[3.3.1. Server-side 6](#_Toc528342624)

[3.3.2. Client-side 6](#_Toc528342625)

[3.4. Security Architecture 6](#_Toc528342626)

[3.5. Communications Architecture 6](#_Toc528342627)

[3.6. Performance 7](#_Toc528342628)

[4. System Design 7](#_Toc528342629)

[4.1. Use Cases 7](#_Toc528342630)

[4.2. Database Design 7](#_Toc528342631)

[4.3. Data Conversions 7](#_Toc528342632)

[4.4. Application Program Interfaces 7](#_Toc528342633)

[4.5. User Interface Design 7](#_Toc528342634)

[4.6. Performance 11](#_Toc528342635)

[Acronyms, Abbreviations and Terms 11](#_Toc528342636)

# 1. Introduction

## 1.1. Purpose of The Product Design Specification Document

This document has been created to give the reader an overall idea of the system required to run the Contractor Contactor Web Application. This document has been created to help me come up with an organised approach to building out the Web Application’s system and architecture. It is to help a developer and the project manager to completing the goal of this project.

# 2. General Overview and Design Guidelines/Approach

## 2.1. Assumptions/Constraints/Standards

For this Web Application to be fully online, running and to provide the great service it intends; there will be a few requirements.

The user of the Web Application will need an internet connection. If it is offline, the user will be informed of such by their own system.

The extent of the system and how much it can offer to the user will be defined by the amount of work that can be put into it by the end of the project.

This Web Application is a straightforward system and will be available for all devices who can attain internet connection. It will be developed for the latest edition of Chrome and on a Windows 10 PC.

This application is intended to be available for any device and screen size and will be highly user friendly.

# 3. Architecture Design

## 3.1. Logical View

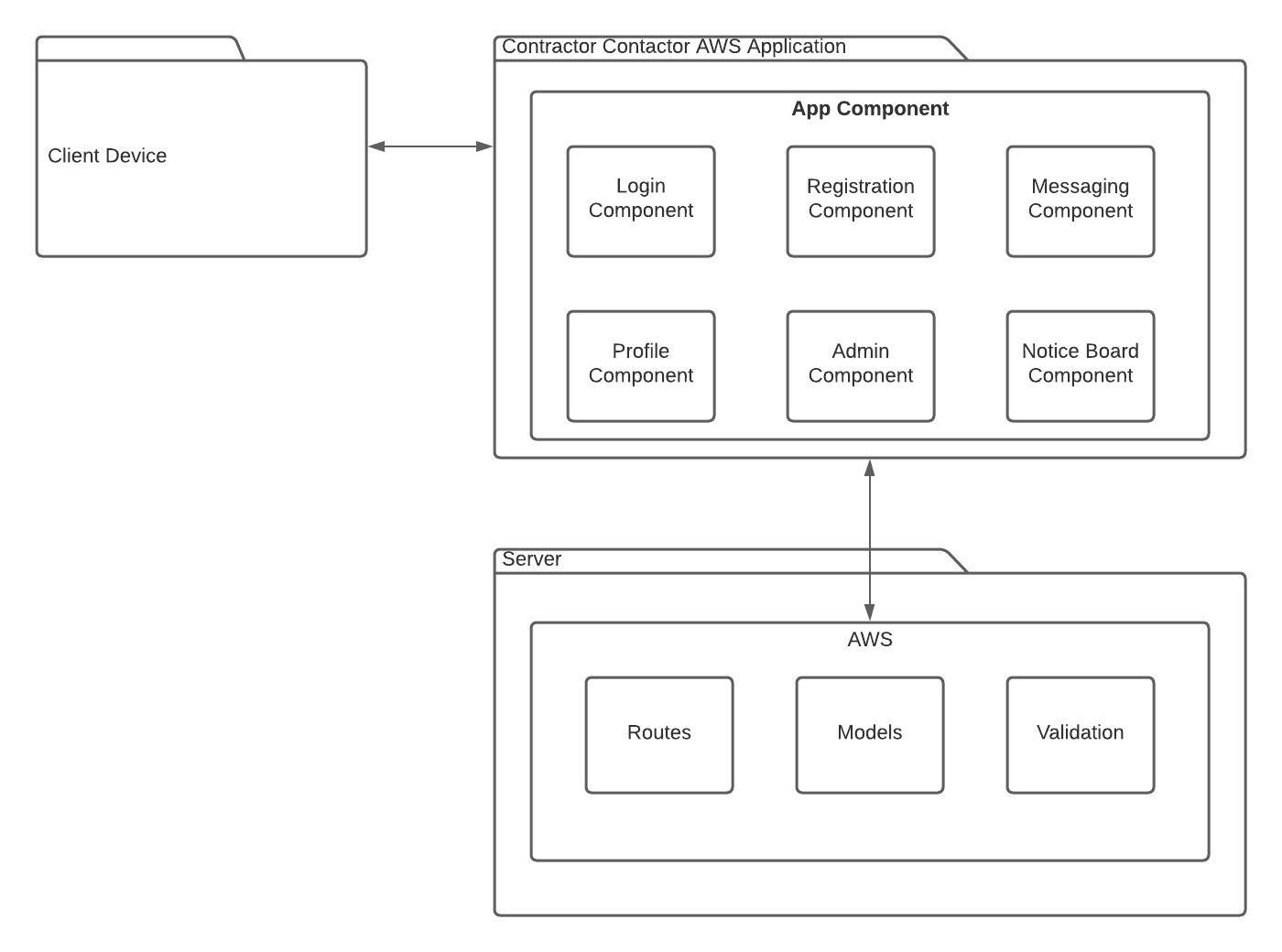


Fig. 3.1.1

In picture above displays the overall system of the Contractor Contactor.

In this representation we start with the user’s browser. They access this on the device that they wish to do so on.

They will get to our application via a URL request. This is a website it will be programmed through ruby, HTML, CSS and JavaScript on AWS. It will be a single page website which functions as a multipage. It will be very quick and very user friendly. As indicated there are several pages which the user can navigate through such as the login and the Profile. The flexibility of the website will be handled by routes and the information will be stored in the models. There will be some validation required such as validation from the Admin of new profiles.

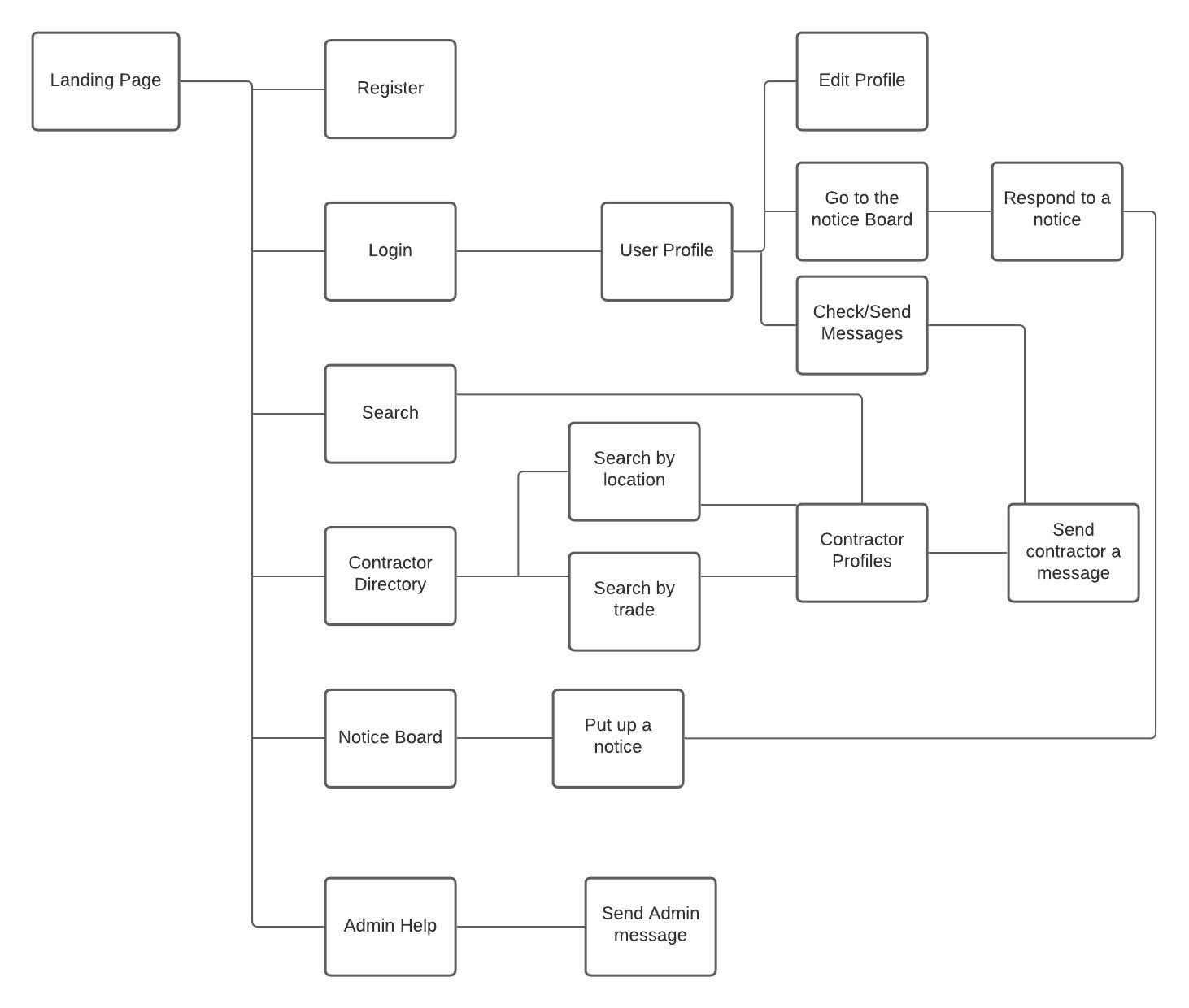


Fig. 3.1.2

This is the site map. We start on the landing page where you have several options. You can register an account, login into your account, search for a type of contractor, enter the contractor directory, view the notice board and enter the admin help section.

Once you log in, you have a few further choices. You can still use the site as previously but you have a profile option and here you can edit your profile if you are a contractor and a customer. Only contractors can go to the notice board, respond to notices, check messages that the customers have sent them and send messages back.

If you search it will bring you into the contractor profiles with similar words as the requested.

You can enter the contractor directory where you may search for a contractor by location or by trade or both. This will bring you a list specific to the choices the user made.

Next a user can enter the notice board and put up a notice. Here they will leave their email to get the response. Contractors can reply to the notice.

Finally we have the admin section. Here you can see specify sections like the FAQs etc. and most importantly you can communicate directly with the admin for better customer support on the app.

# 3.2. Hardware Architecture

**We will be using Heroku. When we have built our website we want universal access to it across the web. Therefore we need a server and a domain. This is why we will implement Heroku. It will be on the Heroku Server.**

**It is easy to set up, easy to use and very reliable. On the Web Application will be available to the user as long as Heroku is available.**

# 3.3. Software Architecture

Contractor Contactor Web Application Software Architecture will use the following software:

## 3.3.1. Server-side

HTML

CSS

JavaScript

## 3.3.2. Client-side

Ruby on Rails

# 3.4. Security Architecture

There will be a profile validation done by the admin to make sure the contractor who wishes to set up a profile is a legitimate person with upstanding intentions.

# 3.5. Communications Architecture

# This Web Application will make use of HTML CSS and some JavaScript on the front end. The user will interact with the front end when using the application. This front end will sent the data to the back end.

# Data will be required, such as login information to access entry to the Web Application, profile information for the contractors to be able to use the site as a contractor with all the benefits that entails.

# The tables in the database in AWS will be planned on the SQL workbench and then scaffolded into the AWS system. Here all the data will be stored for the website.

## 3.6. Performance

AWS and Heroku are very well made systems which provide excellent and efficient service to many users at once. We can expect speed, flexibility and no problems with the functionality with these reputable systems.

# 4. System Design

## 4.1. Use Cases

Use cases for the Contractor Contactor application can be found in the Requirements Specification documentation.

## 4.2. Database Design

At first I will use the SQL workbench to design my database as I am very familiar with this too. I will create all my tables, primary keys, foreign keys and relationships.

After this I will test it out several times to make sure that it runs correctly.

When this is working to an optimal level and I am satisfied that it will serve the Web Application to a high impeccable level I will reassess it for the Ruby on Rails code. After this the code will be scaffolded into the AWS system so that the database can be accessible for the Web Application.

## 4.3. Data Conversions

Data will be moved around the system at high speed. It will go from the front end to the back end to the database, where it is stored for later use. If it is requested it will be sent to the client for review by the user of the Web Application.

The data will be stored in AWS on the tables we have scaffolded in.

## 4.4. Application Program Interfaces

The API is the way for the user to communicate with the Web Application. The user sends a HTTP request to the Web Application, for example to get access to the Contractor Profiles. The Contractor Profile section will receive the request and respond with the correct type of contractor profile.

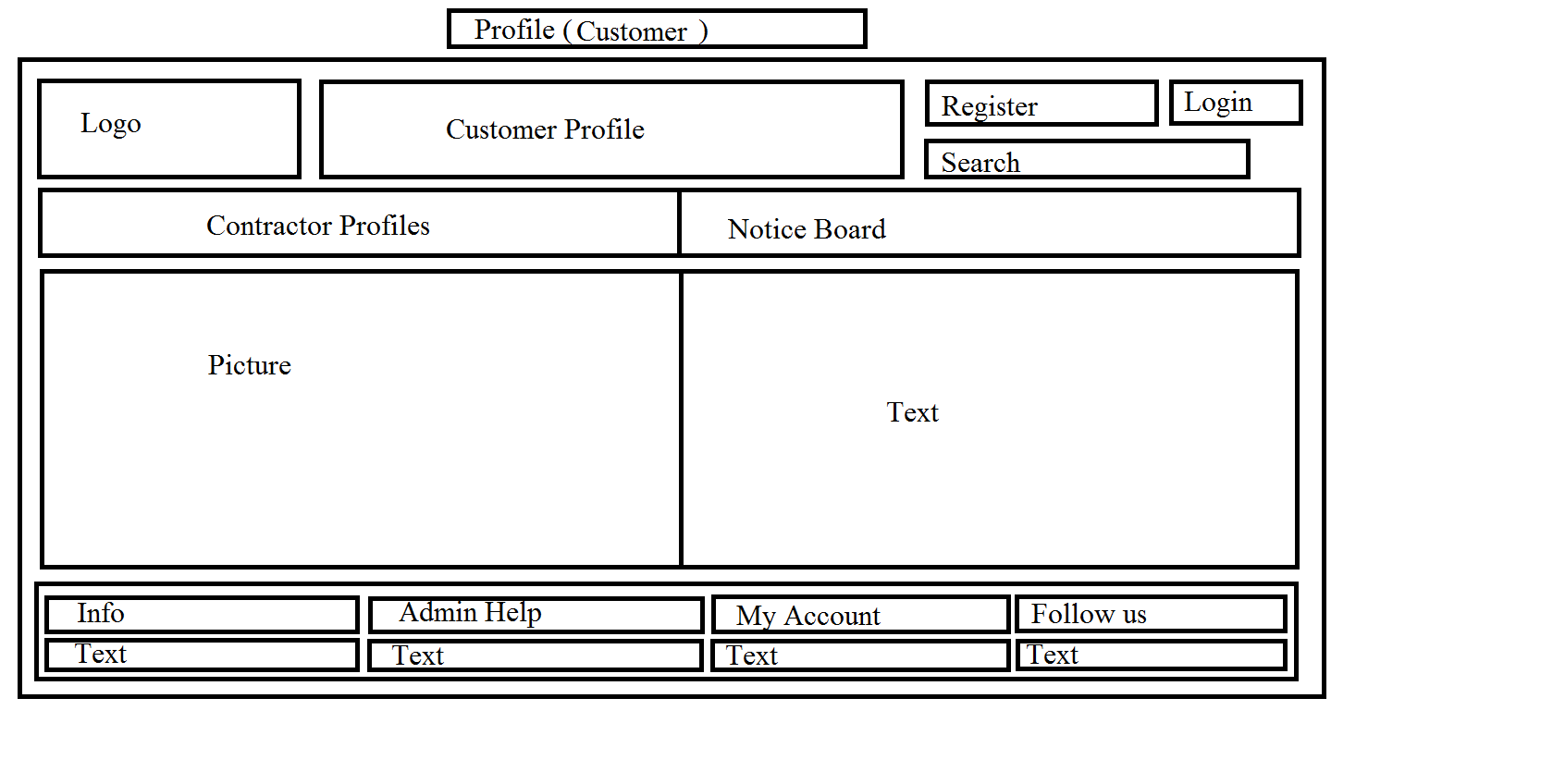
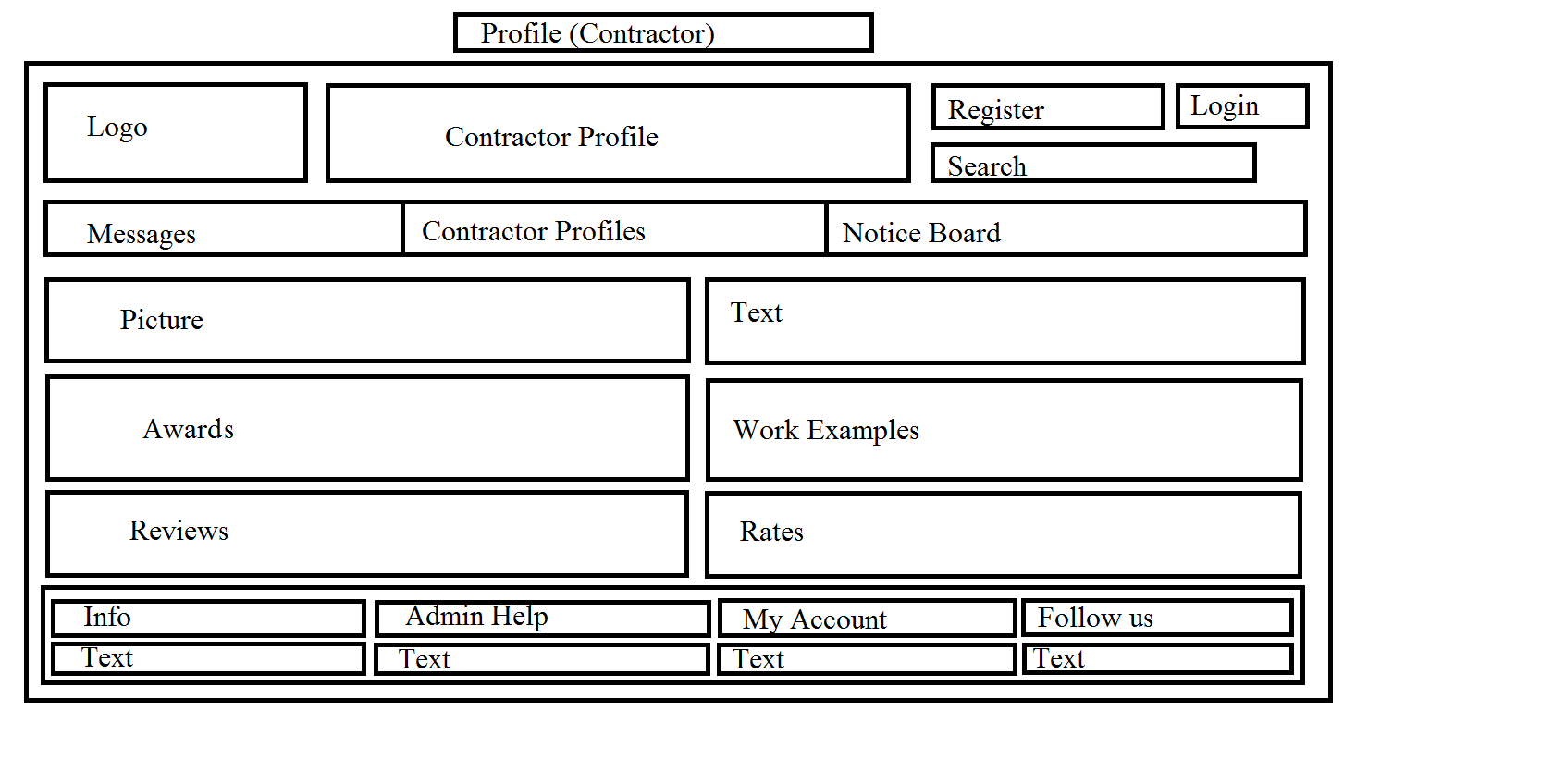
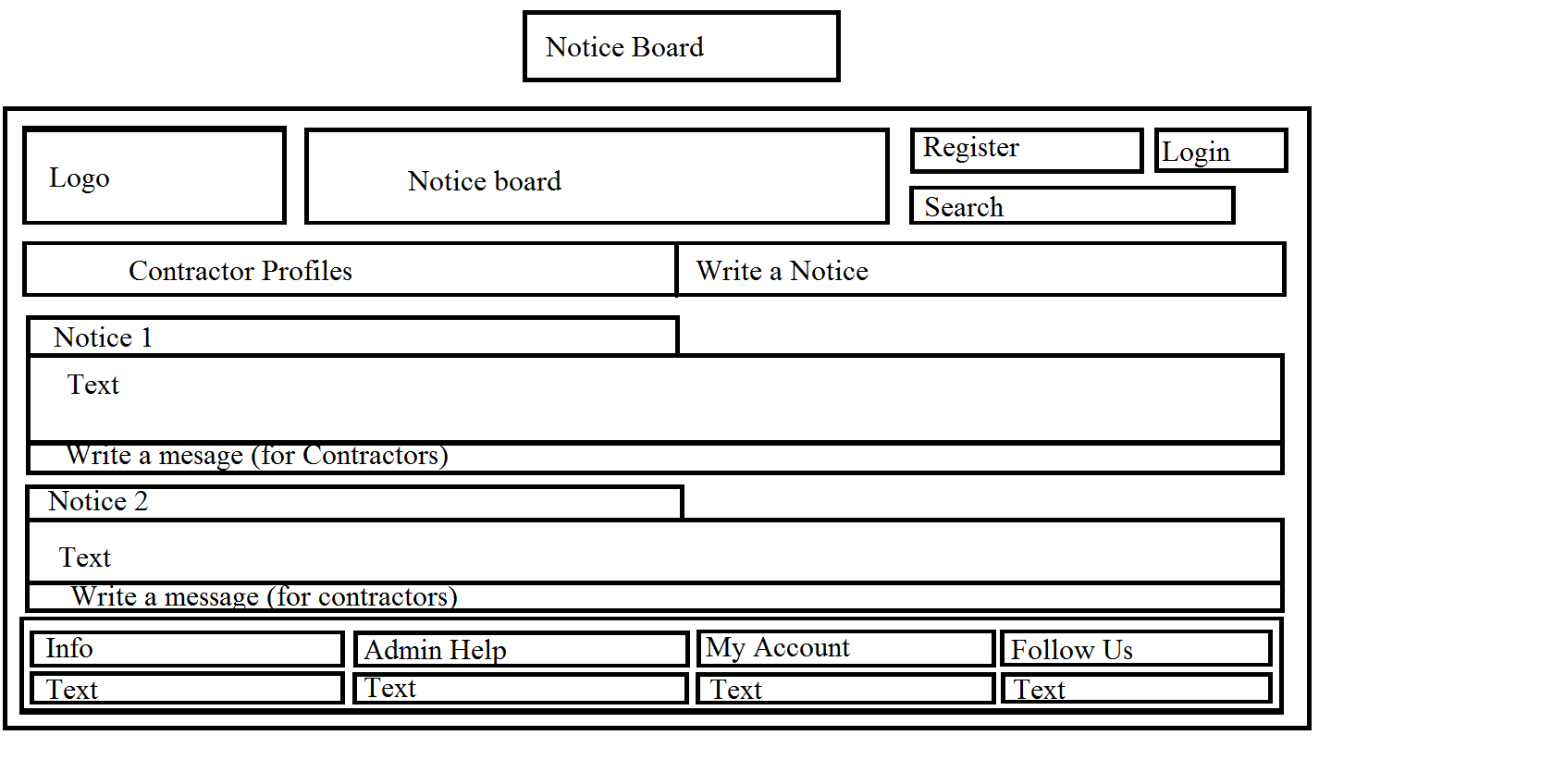
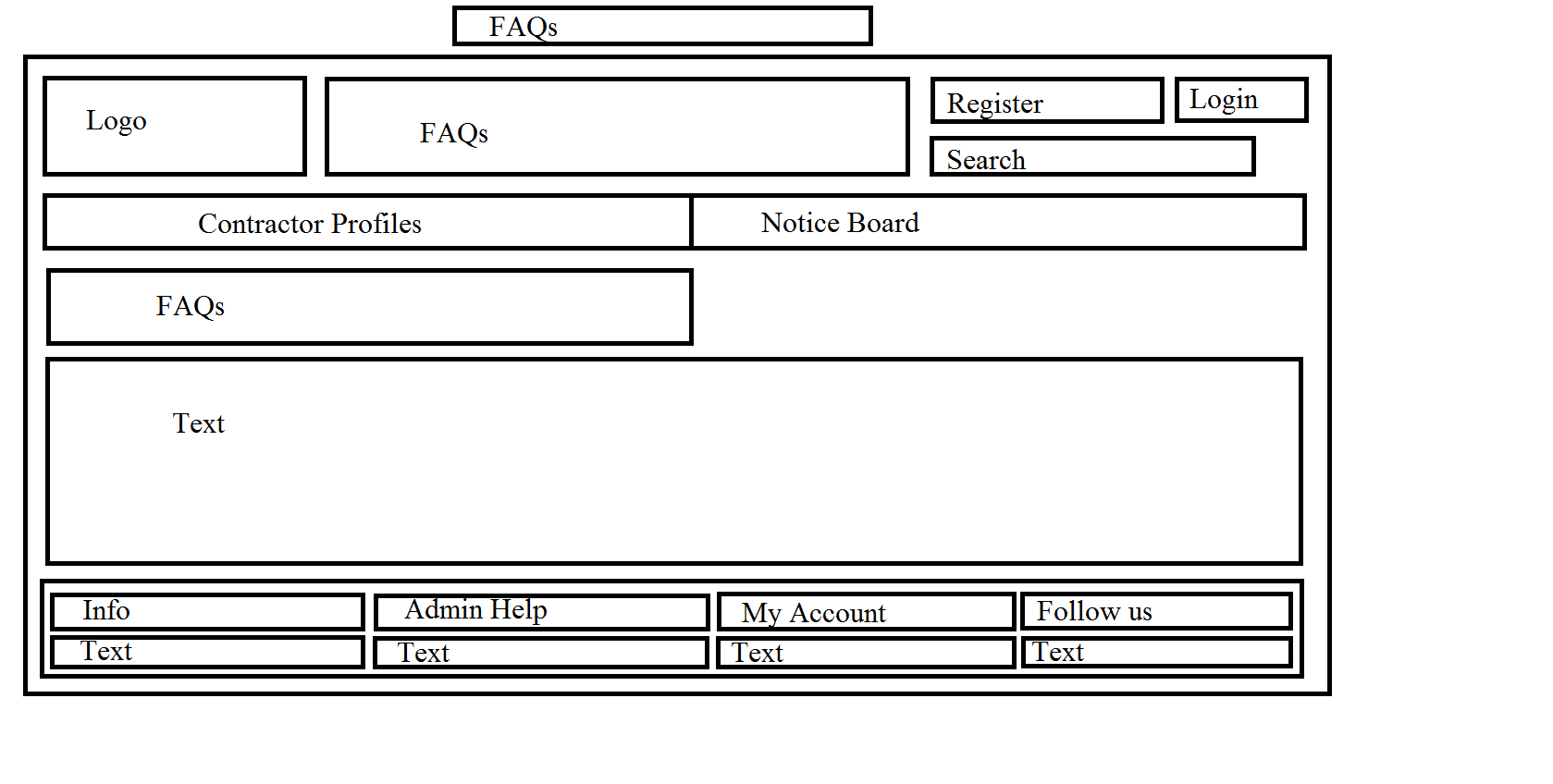
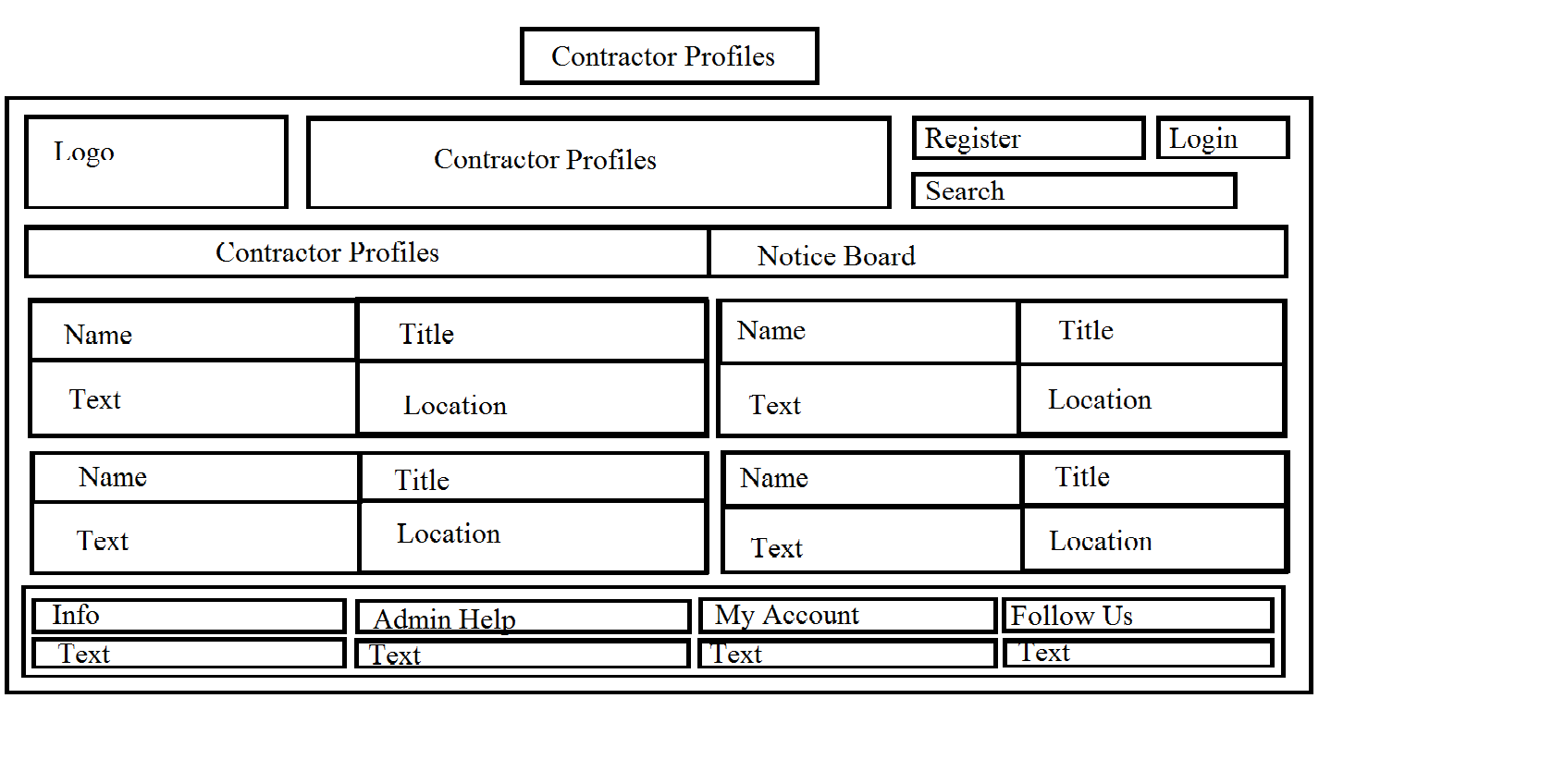
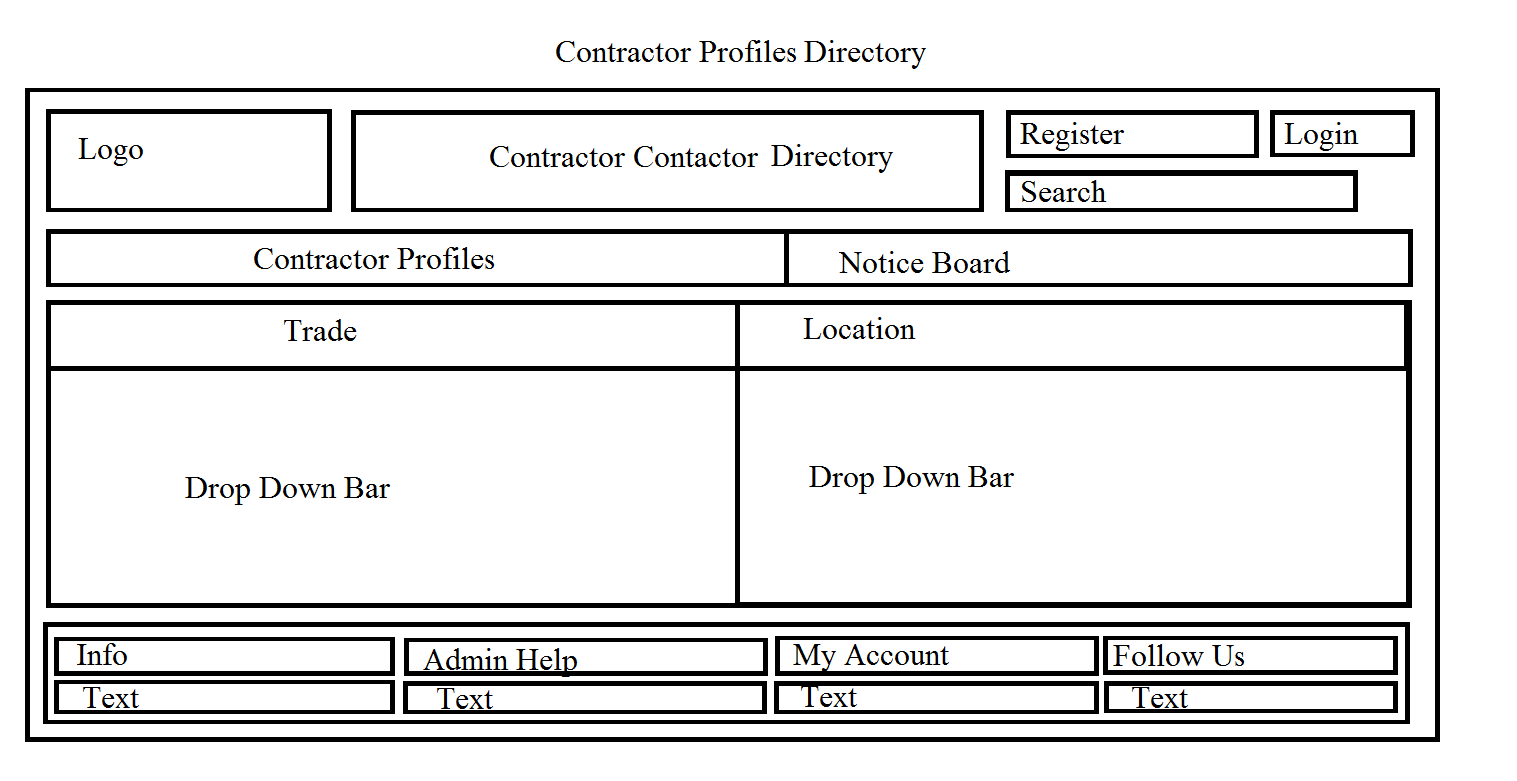
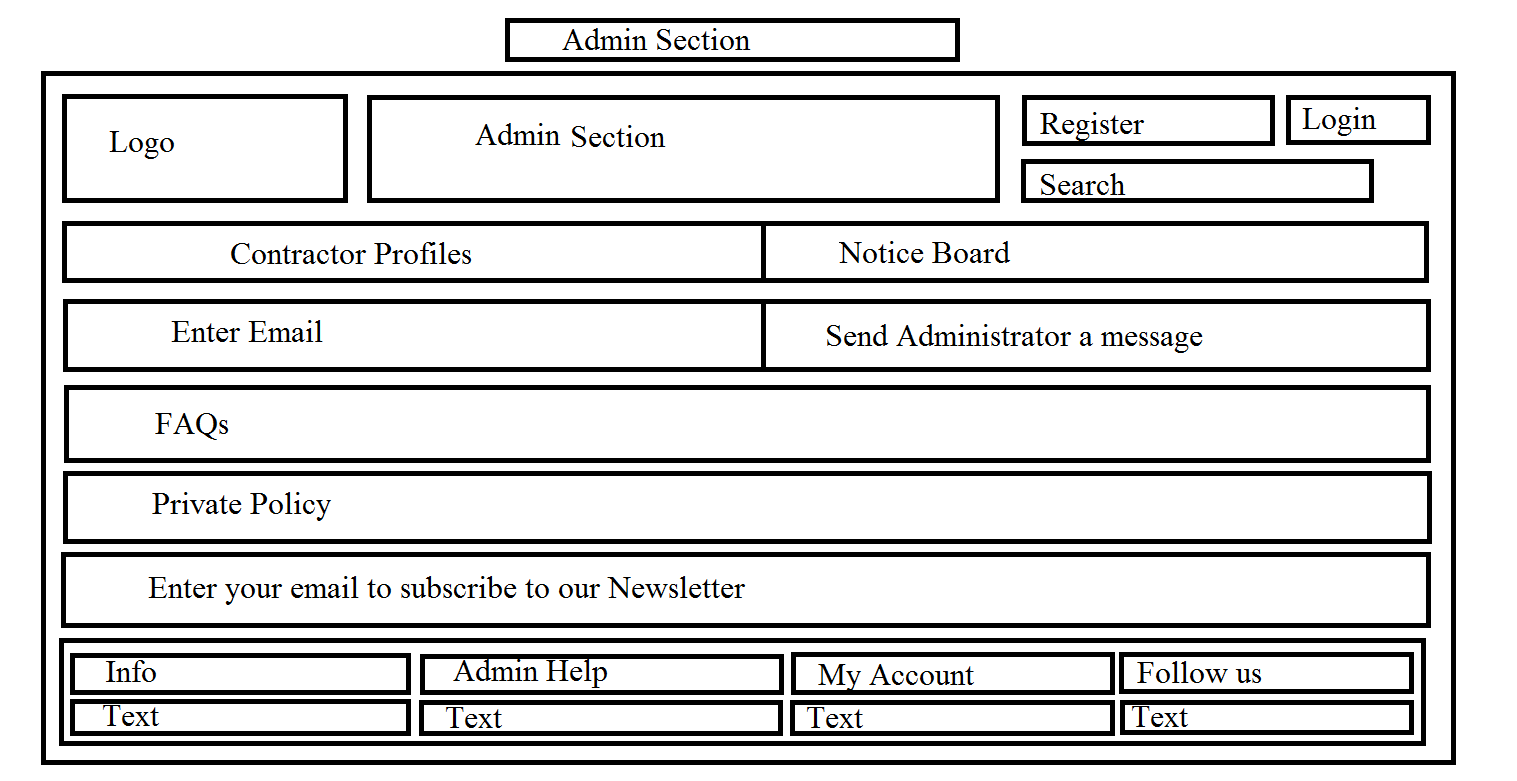
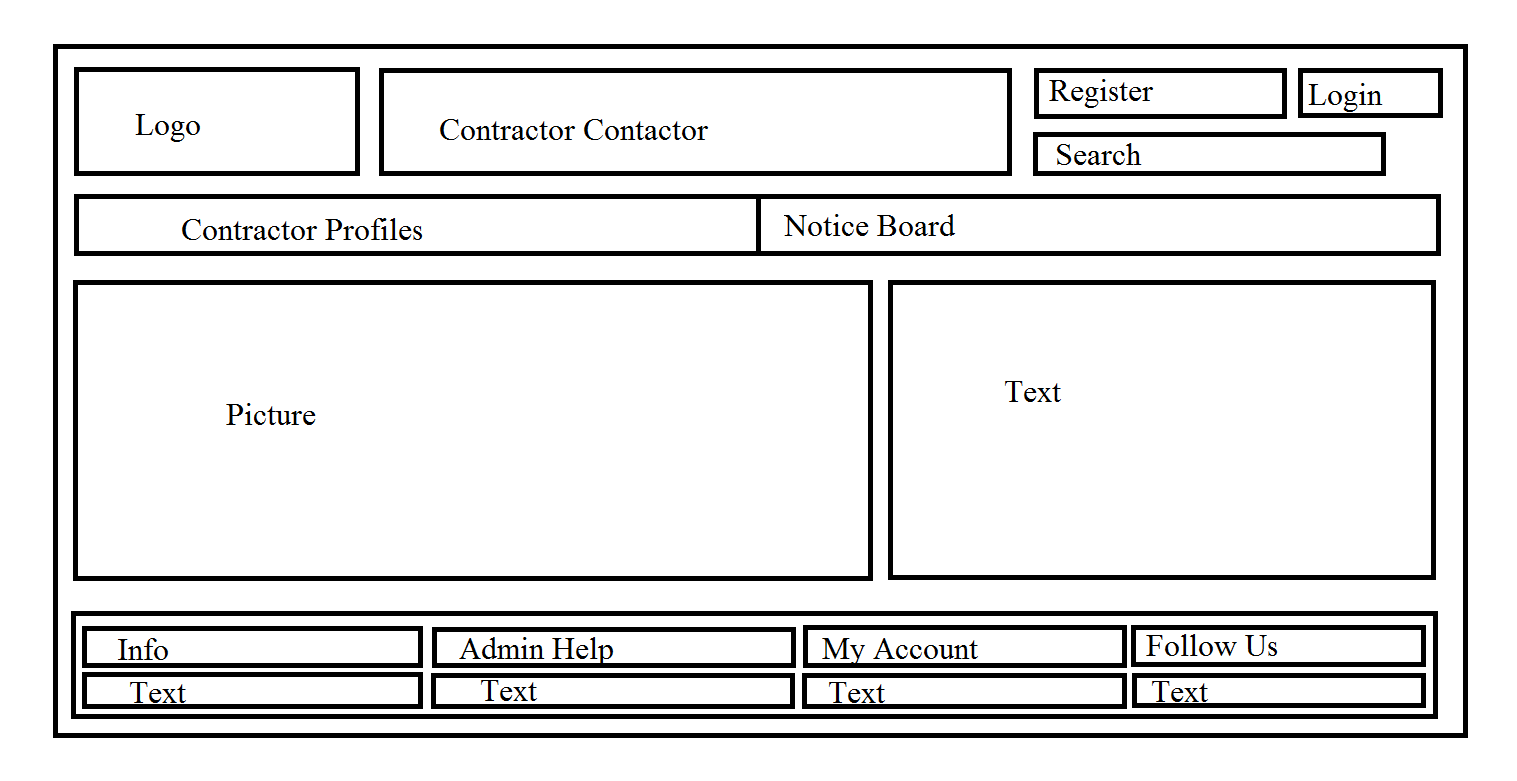
## 4.5. User Interface Design

The User Interface Design will be created with 3 technologies. On the front end we will make a very user friendly interface using HTML, CSS and JavaScript. These technologies shall be written into AWS.

All of these technologies are the core languages for any Web Application. They will create a well-structured website with an enticing aesthetic feel and a high level of responsiveness. All of this will create a unique and enjoyable experience for the user which will encourage them to return.

Here are some wireframe examples of what the user interface will look like:

This is the main page.



## 4.6. Performance

If someone accesses our page they will find it to be very user friendly, aesthetically pleasing as it will work very fast. This is due to the high level of code, the particular coding languages, Heroku and the simplicity of the Website.

The only thing that may hold back this website would be the internet connection of the user, whether Heroku or the Contractor Contactor We Application is undergoing updates.

# Acronyms, Abbreviations and Terms

|  |  |
| --- | --- |
| HTML | Hyper-text Mark-up Language, the language that provides structure to web pages. |
| CSS | Cascading Style Sheets, adds styling to web pages. |
|  |  |