Documentation

**Vinesh Mistry**

**1509170**

An Assignment Submitted in the partial fulfilment for

the award of Bachelor’s degree in Computing

|  |
| --- |
| **May 2018**  ***The University of Bolton***  *Deane Road, Bolton, BL3 5AB*  http://www.bolton.ac.uk |

Contents

Contents 2

List of Figures 2

Technologies used 3

Additional libraries 3

Objectives 4

Objectives Achieved 4

Objectives Not Achieved 4

Changes from the SRS 5

Code Documentation & Inline comments 6

Testing AND Validation 7

Future Development 9

Bibliography 10

**PRESS F9 TO UPDATE THE CONTENTS PAGE (this is non-printing text)**

**PRESS F9 TO UPDATE THE LIST OF TABLES (this is non-printing text)**

List of Figures

Figure 1 - PHPDoc example 6

Figure 2 - Inline comments example 6

Figure 3 - Using Validator Class 7

Figure 4 - Example of HTML type validation 8

Figure 5 - Example of HTML required validation 8

**PRESS F9 TO UPDATE THE LIST OF FIGURES (this is non-printing text)**

# Technologies used

The technologies used for this development section have not changed from the SRS document. The main technologies used were OctoberCMS (PHP framework) and PostgreSQL (object relational database). The Linux and Apache technologies are still used to host the project.

GitHub was used in order to version control the project. The repository can be found at: <https://github.com/VinMistry/WillService>.

## Additional libraries

The when developing the project, it was found that OctoberCMS does not have a large plugin market. In order to create the PDF for the DIY a PHP library called mPDF was used. This library converts the HTML and CSS styling into a PDF document and outputs this as a file allowing the user to download the file.

# Objectives

## Objectives Achieved

1. User can log into the site
2. User can start creating a will
3. User’s will data is stored in the database
4. User can pay for a will
5. User can download a PDF of their will

## Objectives Not Achieved

1. Buy will addons
2. Multi-testator wills
3. Multi-executor wills
4. Multi-beneficiary wills
5. Complete UTP integration

# Changes from the SRS

There were several changes to the database, to allow for the addition of multiple testators, executors and beneficiaries. Therefore, the following tables, were changed: appointed executors, executors, professional executors, beneficiaries, residual estate and will, in order to accommodate for this change.

The design of the forms has changed slightly to accommodate the theme present in the OctoberCMS front end which was chosen by the client.

# Code Documentation & Inline comments

Each component class of the project has PHPDoc comments, which are similar to JavaDoc in which they explain any the return types and any parameters. An example of PHPDoc is shown below.

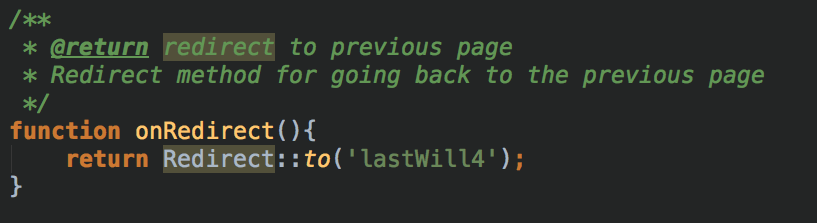


Figure 1 - PHPDoc example

As shown in figure 1 below for the method onRedirect, the method returns a redirection to the previous page. This is shown by the @return part of the PHPDoc. All methods created are documented in this way to allow easier understanding of what a particular method does.

The code also has inline comments, which are displayed using double forward slashes (//) or by using forward slash then a star (/\*). These can be found at various points in the code to show what is happening in a certain section of a method. An example of this is shown below in figure 2.

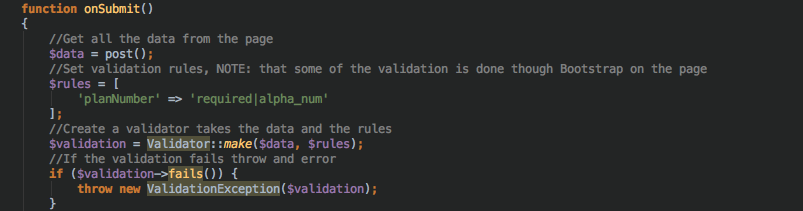


Figure 2 - Inline comments example

# Testing AND Validation

The project was tested via manual testing, this involved inputting data into each page and then checking the database in order to see if the data went into the database. Checking links, pdf downloads and redirects was done via manual testing as well. OctoberCMS has some built in features that allow data to be checked before it is input into the database when using models. This allows incorrect inputs of data to be brought up as an error to the user, this gives an indication as to what went wrong with the input. It also is helpful when debugging code, where column names do not match up.

In order to prevent users from entering values that do not align with the format of the entry, validation rules were made using OctoberCMS’s Validator class. This takes the data and validates it against a set of defined rules. An example of this is shown below in figure 3.

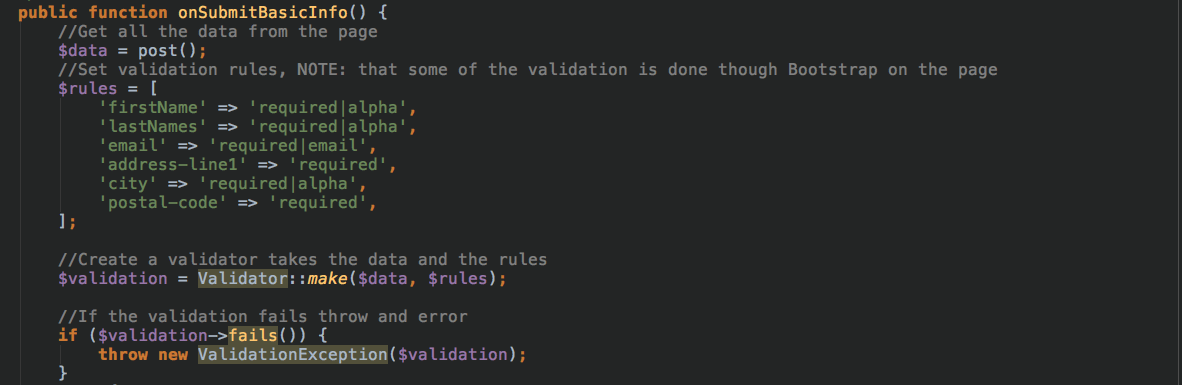


Figure 3 - Using Validator Class

From figure 3 we can see that the rules are specified using a key value pairs, the key being the input’s name value and the value being the rules to apply against the input. This is then used along with the data to create a validation, the Validator then can output whether the validation failed using the fails() method. In order to display the error an exception is thrown using the ValidationException class.

The validator class was implemented at the later stages of the development and therefore there is additional validation in the HTML, using the Bootstrap framework and HTML5 type and required attributes. HTML5 type attribute allows declaration of input types such as number, email and text which checks the formatting of the entered data to ensure that it is correct. For example, if the type is set to email, the input field will display an error if the user enters “testatgmaildotcom” or even “testgmail.com”. This reduces the likelihood that users would enter incorrectly formatted data. An example of the type attribute working is shown below in figure 4.

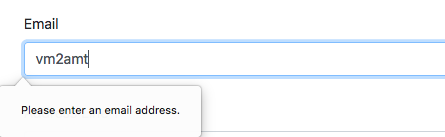


Figure 4 - Example of HTML type validation

The required attribute, allows the form to check whether a field is empty. If it is an error will be displayed and the user will not be able to continue unless they enter a value. An example of the required attribute working is shown below in figure 5.

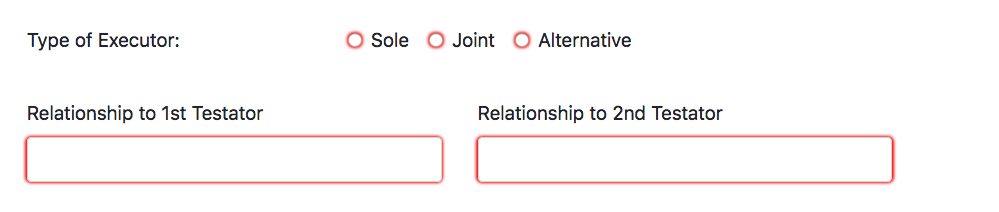


Figure 5 - Example of HTML required validation

# Future Development

The main focus of a future development should be to fix any issues present in code, as well as, implement any features listed in the objectives not achieved section.

After doing so, the development should be revaluated to see where the next need for improvement may be. In addition to this, looking at the SRS for any possible features that need to be implemented for a fully functioning system.

## Grade I think I deserve

I think this assignment deserves an A. The quality of comments and documentation, and the code itself is of a high standard. The web application is successfully deployed using the LAPP stack and provides a good foundation for future developments. The video provided alongside this document shows a comprehensive look at how and what has been developed. Although there is no way to show the manual testing apart from the demonstration in the video, it is apparent that some testing has been done.

The application used MVC in order to display each section, this is the way OctoberCMS is intended to be used. This comes with separation of concerns which neatens the code and allows HTML and PHP to be properly formatted.

Because of this I think this deservers an A grade.

# Bibliography

mPDF, 2018. *mPDF.* [Online]   
Available at: https://github.com/mpdf/mpdf  
[Accessed 20 May 2018].

mPDF, 2018. *mPDF Documentation.* [Online]   
Available at: https://mpdf.github.io/  
[Accessed 20 May 2018].

OctoberCMS, 2018. *OctoberCMS Documentation.* [Online]   
Available at: https://octobercms.com/docs/cms/themes  
[Accessed 1 April 2018].