

# **TEST HARNESS**

Elliot Keen 14059181



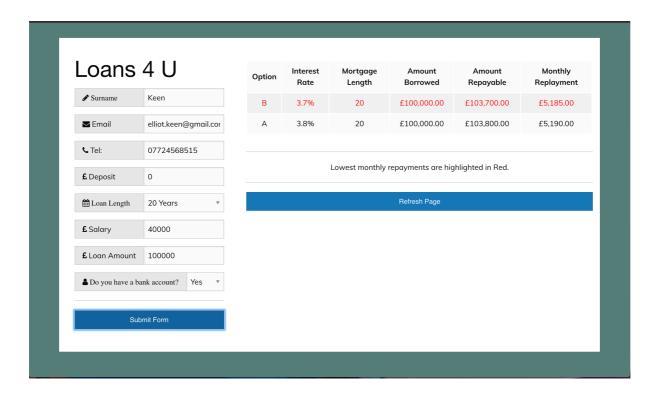
# Introduction:

My loan calculator is shown below. What follows details the testing I have performed on my project, as per the assignment spec.

I utilised the Foundation front-end framework created by Zurb for my front-end styling and formatting. (www.foundation.zurb.com)

I used Font Awesome CSS icons for styling my form. (www.fontawesome.io)

I believe I have achieved a grade of 65% in this assignment having fulfilled all the criteria to reach 60% and highlighted the smallest monthly repayment and ordered the table results accordingly, which is half the criteria for the 60-69% grade bracket. I have displayed my project well using the Foundation framework and presented the information clearly and presented numerous tests within this test-harness document.



#### 40%

Create an HTML form that requires the following information:

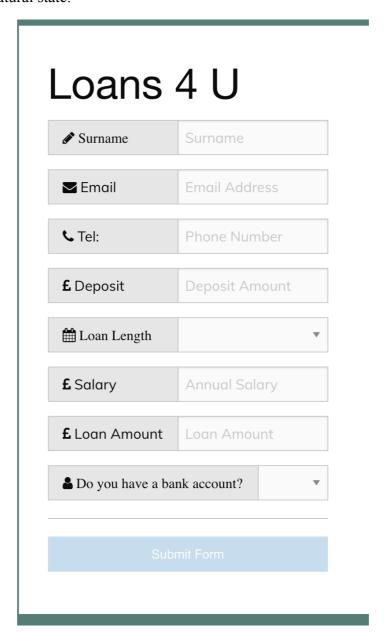
- 1. surname;
- 2. email;
- 3. phone number;
- 4. the size of any deposit;
- 5. the length of time that the user wishes to take out the loan for: allowed values are 10, 20 and 30 years.
- 6. annual salary in pounds;
- 7. how much the user wishes to borrow;
- 8. knowledge of if the user has an account with the building society (as this may allow a preferential deal).

Basic client side data validation is performed on all of the above data, and errors are reported to the client in a simple manner. Details of tests are recorded in test-harness.pdf.

# **Targets Met:**

- Client side data validation is performed on every input
- User is alerted if data is not entered correctly
- Phone number input has custom validation which only allows numbers and a subset of symbols ( '-', '+', '(', ')' ) to be entered into the input box typing any invalid characters does not enter any data into the input box
- Deposit amount, loan amount, and annual salary all include custom validation which only allows numbers to be entered into the input box typing any invalid characters does not enter any data into the input box
- Tests have been recorded and are included in the rest of this section

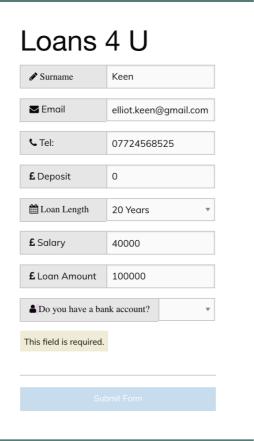
Form shown in natural state:



Form shown without valid data entered (user has clicked through the text areas without entering data). User is unable to submit the form without all valid data entered:



User must enter valid data into every input before submit button is available:



The form validation happens both within the app.component.html and the app.component.ts files. In the app.component.ts file, Angular Validators are utilised to verify the validity of the data entered:

```
constructor(private fb: FormBuilder) {

// Has basic pattern matching to check when number is required if number is given

// Also has validator to check if valid email address given

this.rForm = fb.group({
    'surname' : [null, Validators.required],
    'email' : [null, Validators.compose([ Validators.required, Validators.pattern(/^[0-9]*/) ])],
    'phone_number' : [null, Validators.compose([ Validators.required, Validators.pattern(/^[0-9]*/) ])],
    'sizeOfDeposit' : [null, Validators.compose([ Validators.required, Validators.pattern(/^[0-9]*/) ])],
    'loanLength' : [null, Validators.compose([ Validators.required, Validators.pattern(/^[0-9]*/) ])],
    'annualSalary' : [null, Validators.compose([ Validators.required, Validators.pattern(/^[0-9]*/) ])],
    'ioanAmount' : [null, Validators.required]
})
}
```

Within the app.component.html, an alert located within the app.component.ts is called when invalid data is enter:

```
<div class="alert" *ngIf="!rForm.controls['sizeOfDeposit'].valid && rForm.controls
['sizeOfDeposit'].touched">{{ titleAlert }}</div>
```

The alerts in the app.component.ts:

```
titleAlert:string = 'This field is required.';
emailAlert:string = 'Please enter a valid email.';
phoneAlert:string = 'Please enter a valid 10-digit phone number.'
```

#### **Tests:**

Scenario	Expected Result	Result
Surname box is clicked but	Alert shows stating that	Success
no data is entered.	field is required.	
Email box is clicked but no	Alert shows stating that	Success
data is entered.	valid email is required.	
Phone Number box is	Alert shows stating that	Success
clicked but no data is	valid 10-digit phone number	
entered.	is required.	
Deposit Amount box is	Alert shows stating that	Success
clicked but no data is	field is required.	
entered.		
Loan Length selection box is	Alert shows stating that	Success
clicked but no data is	field is required.	
chosen.		
Salary box is clicked but no	Alert shows stating that	Success
data is entered.	field is required.	

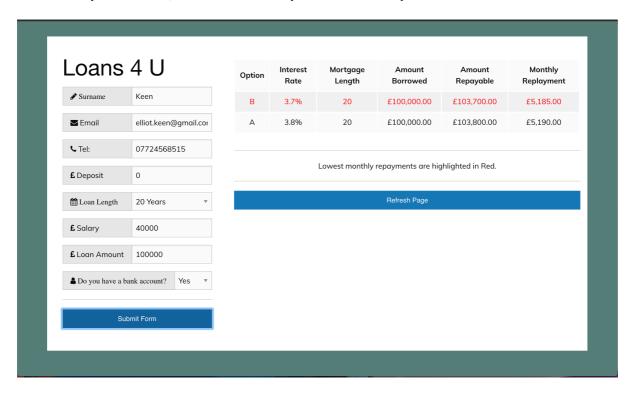
Loan Amount box is clicked but no data is entered.	Alert shows stating that field is required.	Success
Do you have a bank account is clicked but no data is chosen	Alert shows stating that field is required.	Success
Text is entered into the email box but it is not a valid email	Alert shows stating that valid email is required.	Success
Valid email address is entered into the email box.	Alert does not show.	Success
User tried to enter text into phone number box	User is unable to enter text into phone number box, only numbers or '-' or '+' or '(' or ')' are allowed.  Alert does not show.	Success
User enters valid phone number into phone number box.	Alert does not show.  User can enter numbers freely.	Success
User tried to enter text into deposit amount box	User is unable to enter text into Deposit Box, only numbers are allowed.  Alert does not show.	Success
User enters valid numbers into deposit amount box.	Alert does not show.  User can enter numbers freely.	Success

## **Test Summary:**

All validation happens correctly. The user is unable to enter characters within the deposit amount, annual salary, loan amount, and phone number boxes (excluding '-', '+', '(', ')' which are allowed in the phone number box). Angular Validators are used to determine if the email address is a valid email address – that is, if it contains characters separated by an '@' symbol and a '.' and is allowed to contain letters and numbers.

# 40-49%

If and only if the data is correct, then the possible mortgage offers are displayed, these should be displayed at the bottom of the form. So that the user can (for example) modify their deposit value and see the effect on the mortgage offers instantly. At this level, the software need only state the allowed options.



# **Targets Met:**

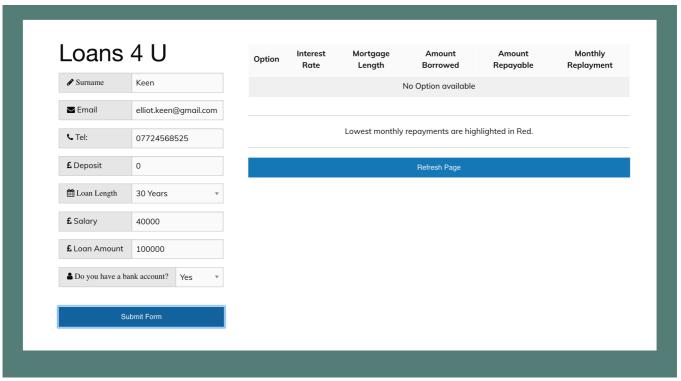
- Offers are displayed on the same page
- Offers can be updated in real time
- Software states all allowed options
- When no options are available, user is alerted to that fact

#### **Tests**

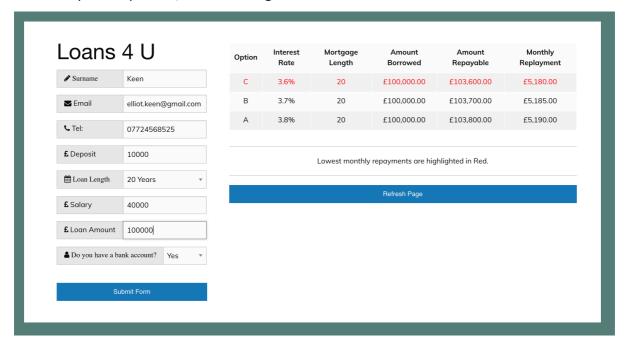
Scenario	Expected Result	Result
20 year mortgage	Options A and B show	Success
<ul> <li>no deposit</li> </ul>		
<ul><li>salary of £40000</li></ul>	Options C, D and E are not	
<ul> <li>holds an account with</li> </ul>	shown	
the building society		
<ul> <li>loan amount £100000</li> </ul>		
<ul> <li>10 year mortgage</li> </ul>	Options A and B show	Success
<ul> <li>no deposit</li> </ul>		
<ul><li>salary of £40000</li></ul>	Options C, D and E are not	
<ul> <li>holds an account with</li> </ul>	shown	
the building society		
<ul> <li>loan amount £100000</li> </ul>		

30 year mortgage	No options show	Success
	No options snow	Success
• no deposit		
• salary of £40000	Error message is shown	
holds an account with		
the building society		
loan amount £100000		
<ul> <li>20 year mortgage</li> </ul>	Option A, Option B and	Success
<ul> <li>deposit of £10000</li> </ul>	Option C show	
<ul><li>salary of £40000</li></ul>		
<ul> <li>holds an account with</li> </ul>	Options D and E are not	
the building society	shown	
<ul> <li>loan amount £100000</li> </ul>	5.1.5	
30 year mortgage	First, no options are shown.	Success
<ul> <li>no deposit</li> </ul>		
<ul> <li>salary of £40000</li> </ul>	Once data is updated,	
<ul> <li>holds an account with</li> </ul>	Option A, B and C are shown	
the building society	option 7, b and c are snown	
• loan amount £100000	On so data is undated	
	Once data is updated,	
Data is changed after result	Options D and E are not	
given to	shown	
B		
20 year mortgage		
<ul> <li>deposit of £10000</li> </ul>		
•		
<ul><li>salary of £40000</li></ul>		

An Example of the error message displayed when no option is available:



#### An example of Option A, B and C being shown:

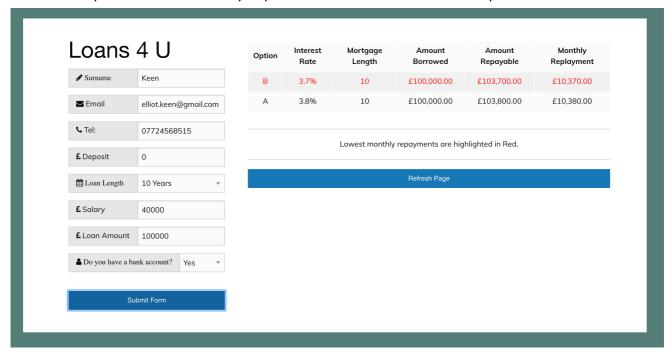


# 50-59%

- 1. At this level Option A and Option B needs to be implemented correctly. Tests should be performed and documented in test-harness.pdf
- 2. The allowed options should be displayed in an intuitive HTML table style, for example with the data in section 0.1.2.1 then the user should see some- thing similar to fig. 1.

# **Targets Met:**

- Options are displayed in an intuitive html table
- Options A and B are fully implemented and tests are shown in previous table



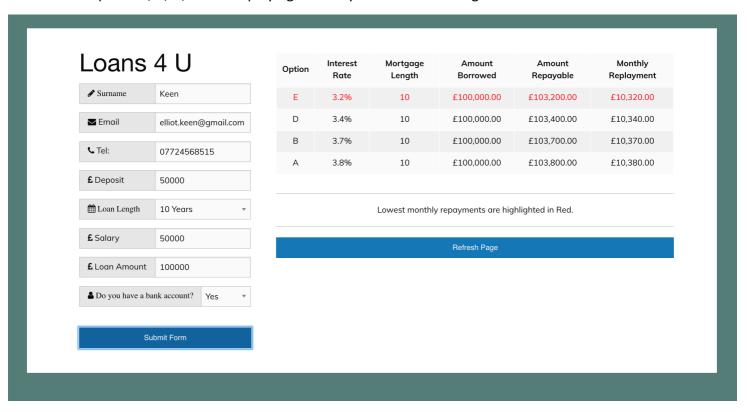
#### 60-69%

- 1. All options A, B, C, D, and E need to be implemented correctly. Tests should be performed and documented in test-harness.pdf. 2. The allowed options should be displayed in an intuitive HTML table style, for example with the data in section 0.1.2.1 then the user should see some-thing similar to fig. 1.
- 2. The form is dynamic in the following manner: if the software detects an error in a field then it will not allow data entry into the form elements after it until the issue in the field is addressed.
- 3. If the data is correct, then the allowed options are displayed and ordered in such a way that the option with the smallest monthly repayment is displayed first in the table and is emphasised in some manner.

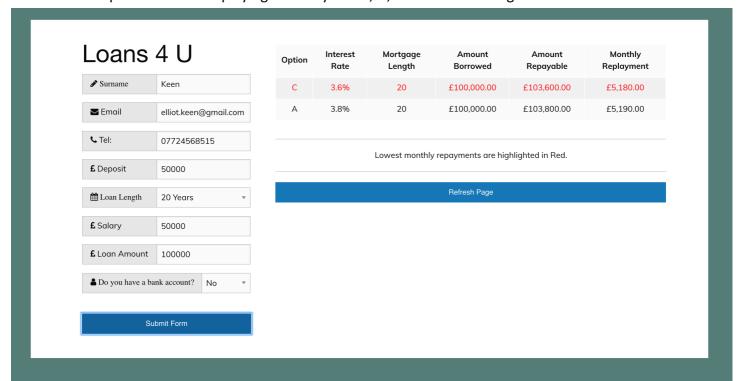
# **Targets Met:**

- All options A, B, C, D and E are implemented correctly
- Tests have been performed showing the proper implementation of options A, B, C, D, and E
- When correct data is entered, allowed options are displayed and ordered in such a
  way as to emphasise the option with the smallest monthly repayment (which is
  coloured in red and displayed first within the table). Remaining options are ordered
  from lowest to highest by monthly repayments

Example of A, B, D, and E displaying correctly and C not showing:



# Example of A and C displaying correctly and B, D, and E not showing:



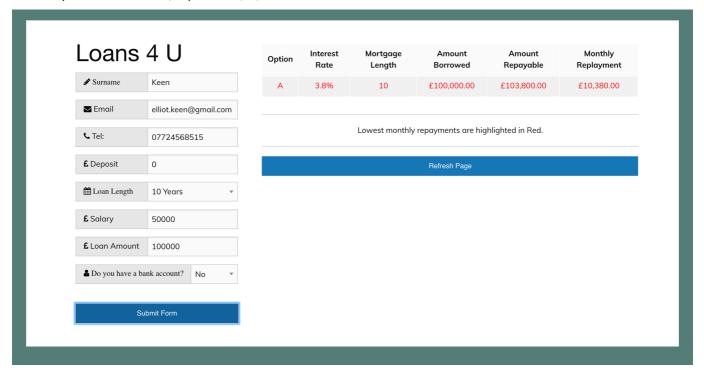
## **Tests**

Scenario	Expected Result	Result
<ul><li>20 year mortgage</li><li>£50000 deposit</li></ul>	Options C and A show	Success
<ul><li>salary of £50000</li><li>does not have building society account</li></ul>	Options B, D and E are not shown	
• loan amount £100000	Smallest monthly repayment is highlighted in red and displayed first	
20 year mortgage	Options A, B, C and E show	Success
• £50000 deposit		
• salary of £50000	Option D is not shown	
<ul> <li>has a building society account</li> </ul>		
loan amount £100000		
30 year mortgage	Option D shows	Success
• £50000 deposit		
• salary of £50000	Options A, B, D and E are	
<ul> <li>has a building society account</li> </ul>	now shown	
loan amount £100000		
<ul> <li>10 year mortgage</li> </ul>	Options A, B, D and E show	Success
<ul> <li>£50000 deposit</li> </ul>		
• salary of £50000	Option C is not shown	

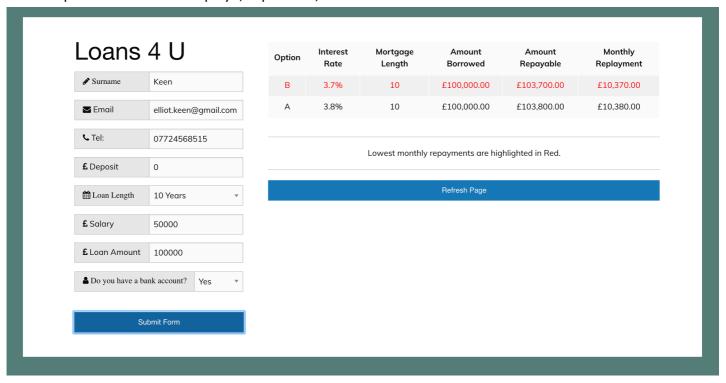
<ul> <li>has a building society</li> </ul>	
account	
<ul> <li>loan amount £100000</li> </ul>	
	Success

# Further tests displaying correct data for every option:

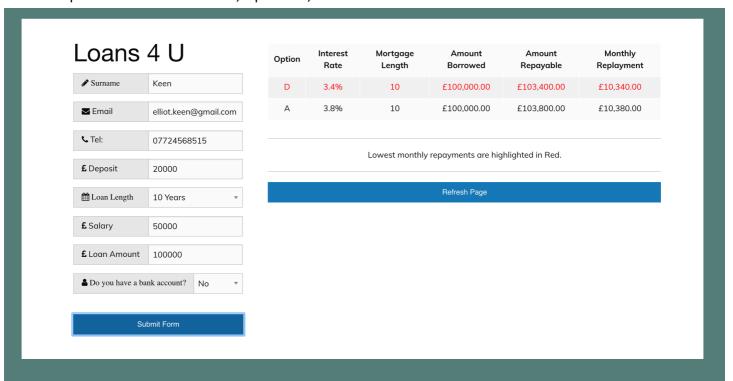
Option A is shown; Options B, C, D and E are not show:



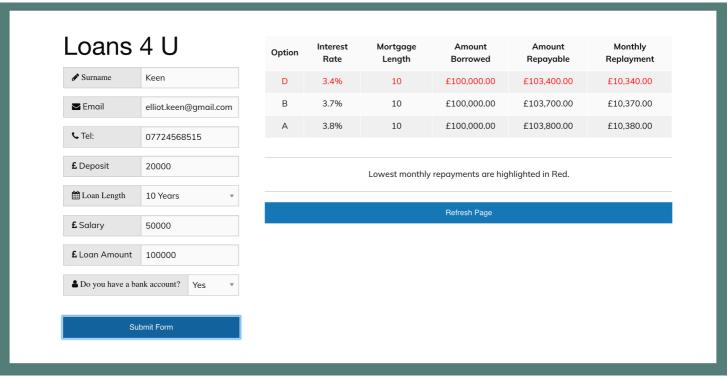
Options A and B are displaye; Options D, C and E are not shown:



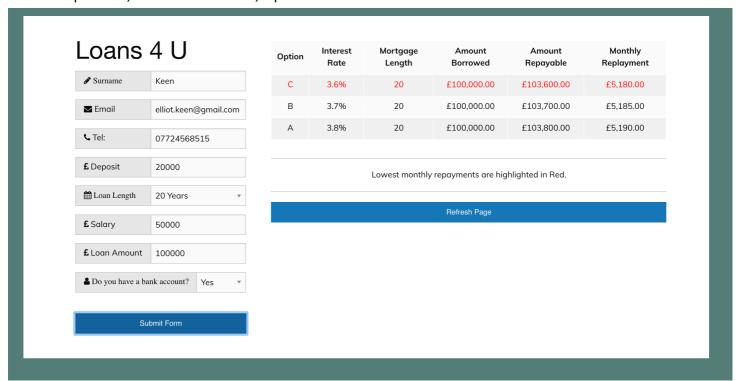
## Options A and D are shown; Options B, C and E are not shown:



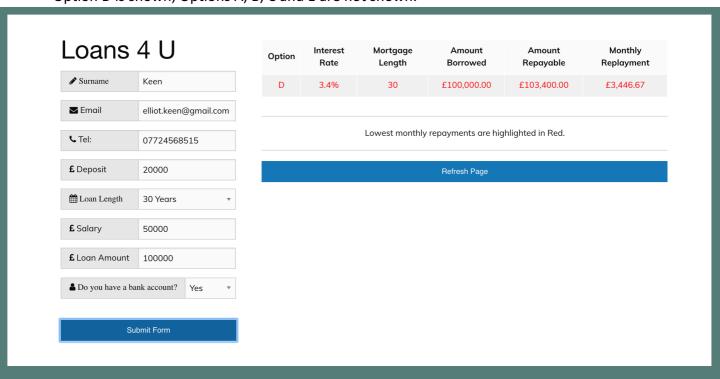
## Options A, B and D are show; Options C and E are not shown:



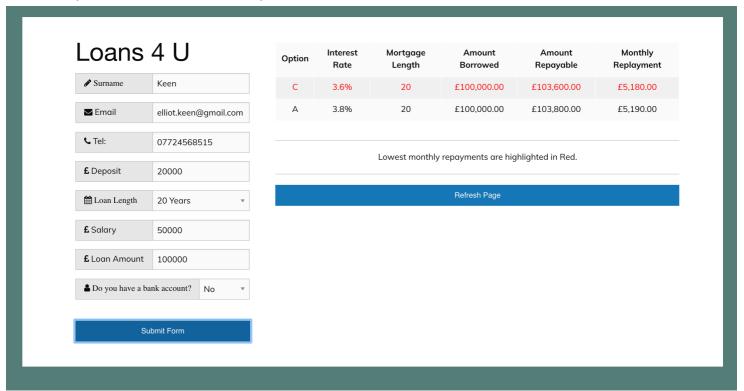
Options A, B and C are shown; Options D and E are not shown:



## Option D is shown; Options A, B, C and E are not shown:



Options A and C are shown; Options B, D and E are not shown:



Options A, B, C and E are shown; Option D is not show:

