Online diary system

Coursework by UP719931

# 1. Introduction

To develop a system which is able to hold people’s diaries consisting of all appointments and accessible through a website. Allow users to have individual access, create appointments and include other users to an appointment. All appointments should be saved in a database. Furthermore, the quality of implementation should be of high standard following the industry’s best practices as seen in lectures.

## 1.1 Future proofing

Although the initial development rationale is to build the system for desktop systems, it should be implemented using CSS responsive design techniques to ensure the system’s user experience on mobile is acceptable as well.

## 1.2 Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Type** | **Status** |
| **2** **Data requirements** |  |  |
| **2.1** **Internally sourced data** |  |  |
| 1. The system must be capable of holding information on: |  |  |
| a) users | M1 | done |
| b) appointments | M1 | done |
| c) admin | D | unfinished |
| 2. The system must maintain relationships between the above; specifically: |  |  |
| a) what appointments a user has | M1 | done |
| 3. Add dummy data to the database. This needs to hold: | M1/2 |  |
| a) create a number of users and appointments to test different scenarios | M1/2 | done |
| b) create an admin user | D | unfinished |
| **3** **Shared online contact list** |  |  |
| 1. Authenticated users should be able to access a list of all contacts. A real name(first name and last name), phone number, email and address should be presented. The results should be presented in alphabetical order. | M1 | done |
| 2. Authenticated users should be able to search through the available records. The results should be presented in alphabetical order. | M1 | done |
| 3. Unauthenticated users should not be able to access any information | M1 | done |
| 4. The system should redirect not-signed in users to login page | D | done |
| **4** **Authentication** |  |  |
| 1. A user should be able to create new user account providing the information in 4.3 if the data is validated successfully | M1 | done |
| 2. The system should indicate if there is an error so the user can further amend it | M1 | done |
| 3. User details should be validated prior to creation and modification: |  |  |
| a) Username should be unique amongst all users and should be at least 6 alphanumeric characters | M1 | done |
| b) Password should never be displayed in the system | M1 | done |
| c) Firstname and lastname are required fields | M1 | done |
| d) Phone is not a required field. However, if presented, should be validated and be presented in those specific formats:  +44 (23) 92846438, (023) 92846438 | M1 | done |
| e) Email is a required field which should be validated towards a common pattern |  | done |
| f) Password and repeat password should be identical otherwise a message should be returned to the message on creation | D | done |
| 4. The system should save information about the creator of the user if any. | M1 | unfinished |
| 5. A user can be modified only by its creator or themself. | M2 | unfinished |
| **5** **Online diary facility** |  |  |
| 1. Each diary appointment should have: |  |  |
| a) A date/time at which the appointment starts | M2 | done |
| c) A textual title of the appointment | M2 | done |
| d) The person who created the appointment (the "owner") | M2 | done |
| e) A set of people who are involved in that appointment (which may or may not include the owner) | M2 | done |
| 2. Clash with any other appointment for any of the people involved: |  |  |
| a) Should be prevented | M2 | unfinished |
| b) A message should be returned to the user creating appointment with information which participant can’t make the appointment | M2 | unfinished |
| 3. Request reports showing the details held by the system: |  |  |
| a) All appointments associated with a named user (the logged in user by default) | M2 | done |
| b) All appointments in the system on a specified date | M2 | unfinished |
| i) A creator of an appointment can cancel one which should delete the appointment | M2 | done |
| ii) A "calendar" view of appointments (i.e. in a grid) is desirable but not mandatory. | D | unfinished |
| **6** **Supplementary functionality** |  |  |
| 1. Create admin capabilities to delete users and appointments if found inappropriate by Responsible owner | D | unfinished |

# 2. Design

## 2.1 Wireframes

Wireframes have been built using Balsamiq to create an overview of the system in a quick succession. They would be further refined in the High-fidelity prototypes. Link to wireframes: <https://drive.google.com/drive/folders/1V7KU8aGLjfSY9MAtKgIe3Gp2RWdpMB0b?usp=sharing>

## 2.1.1 Homepage

The homepage of the system(Fig. 1) would consist of your personal reports, form to create an appointment and a section for all contacts. The page should be secured with middleware accessible only to logged-in users(that should be true for the all other pages of the system apart from login and register pages).

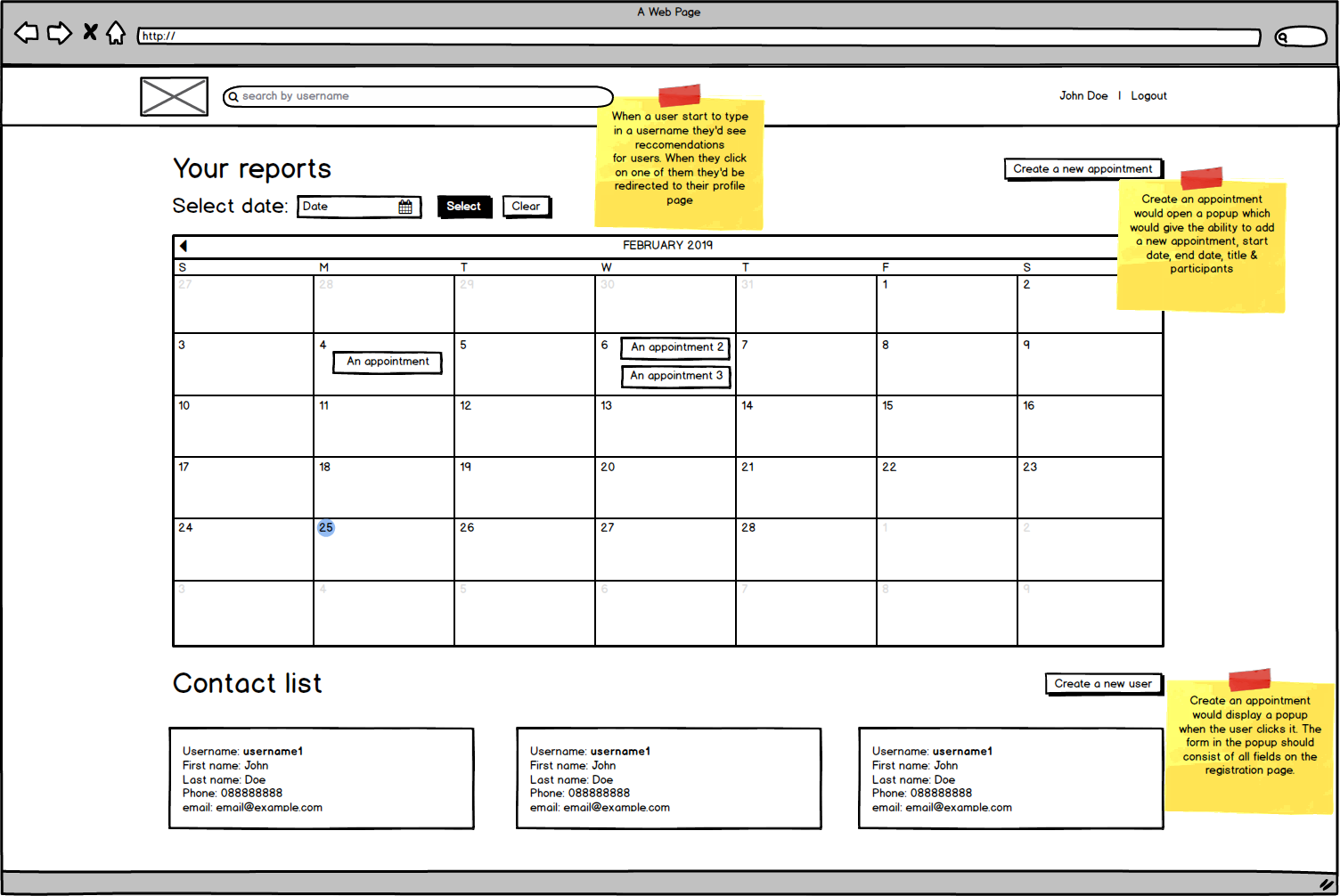


Fig.1 Homepage mockup

## 2.1.2 Login page

Login page(Fig. 2) would be very minimal with the purpose of the user to sign-in or navigate to the registration page.

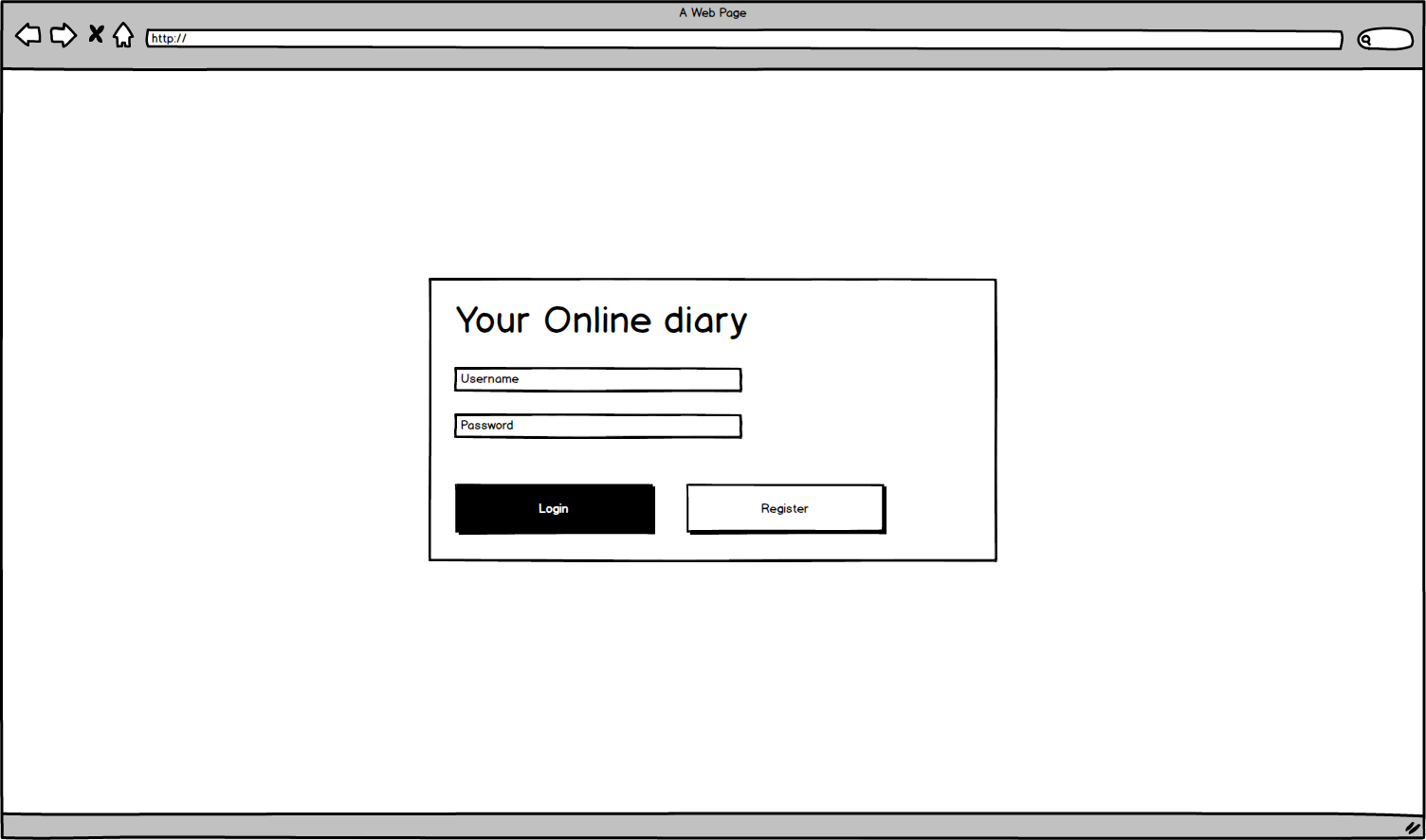


Fig. 2 Login page

## 2.1.3 Registration page

The user interface of the registration page(Fig. 3) would be very similar to the one on the login page contributing to a consistent feel and a better user experience.

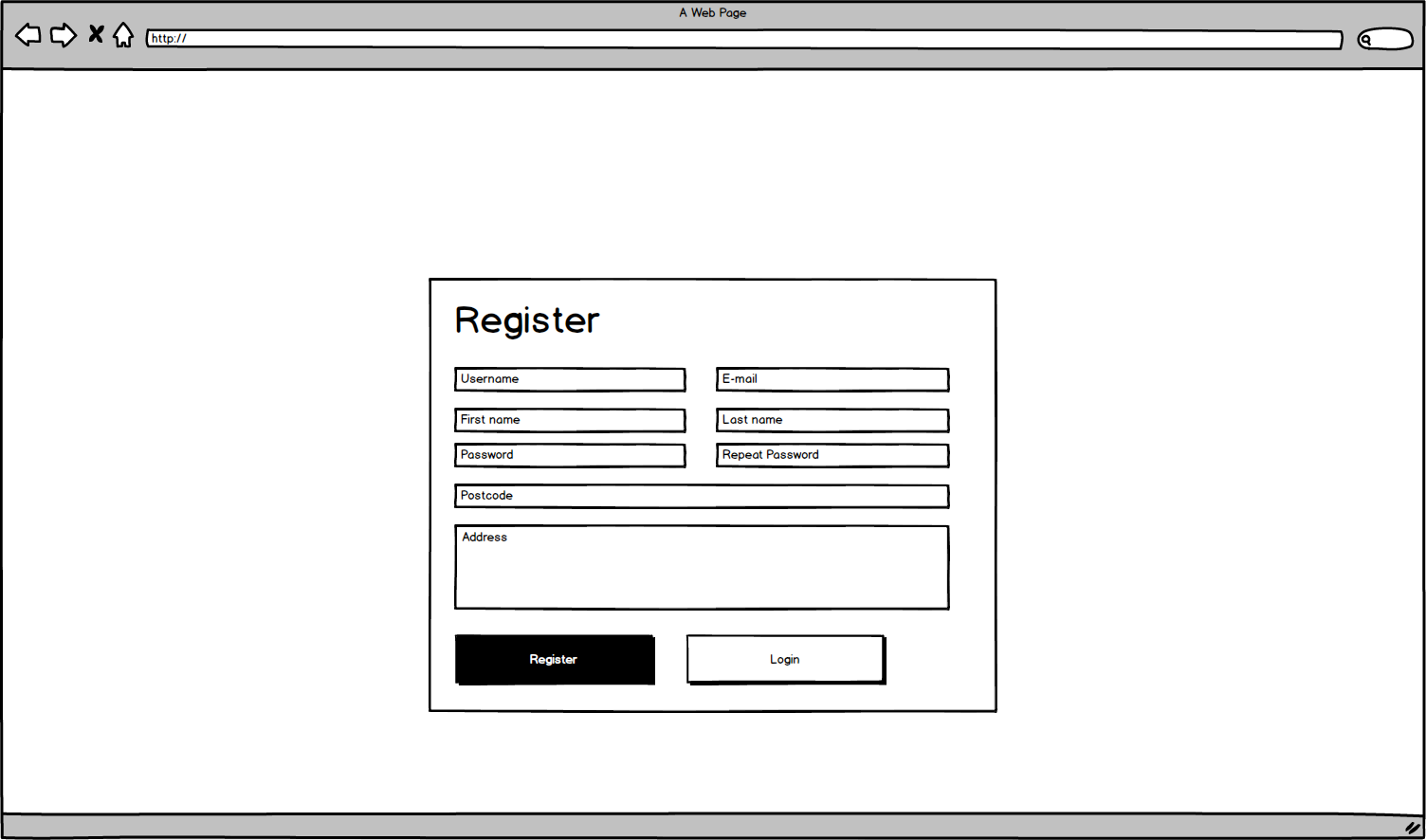


Fig. 3 Registration page

## 2.1.4 Profile page

The profile page(Fig. 4) would consist of a section with User details with a call to action buttons to call and email them. It will also have a list of their appointments using a calendar user interface pattern. It will have a form to allow for a specific date selection which once modified and submitted should change the view to a preview only a specific day and the person’s appointments.

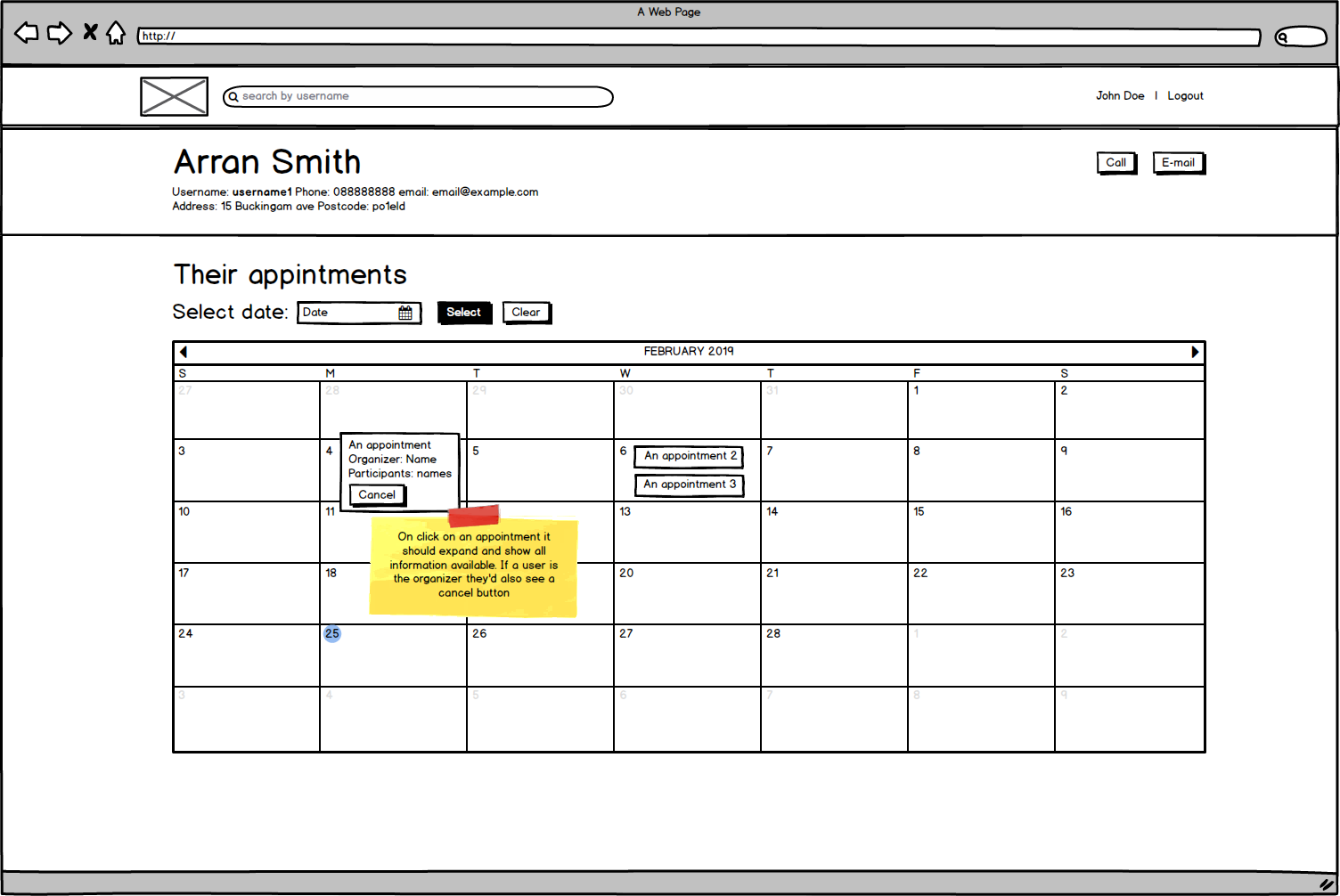


Fig. 4 Profile page

## 2.2 High-fidelity prototypes

High-fidelity prototypes have been built using the Adobe Photoshop software in order to provide a closer feel to the final website(Fig. 5 - Fig. 10). Link to prototypes: <https://drive.google.com/drive/folders/1CKL2dGqeMh2fyxUJGrw1_sCtYYRMYvb4?usp=sharing>

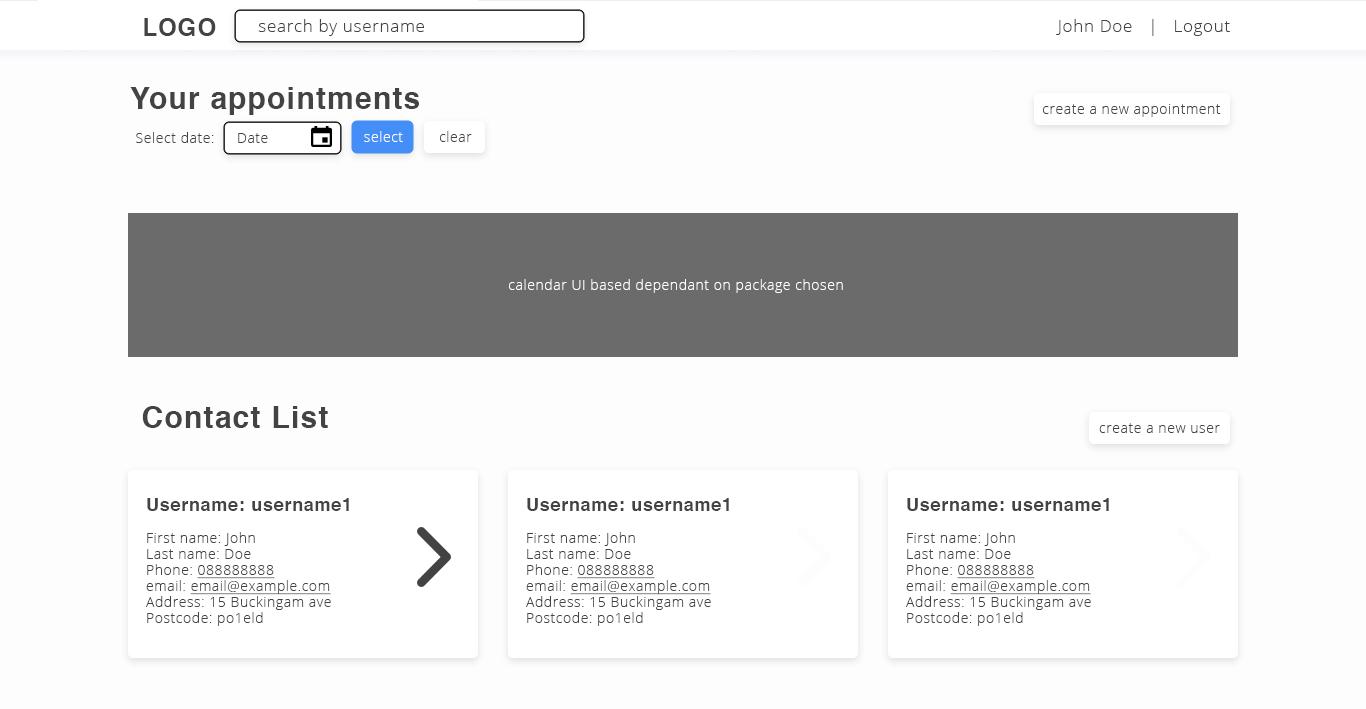


Fig. 5 Homepage user interface with a missing calendar element

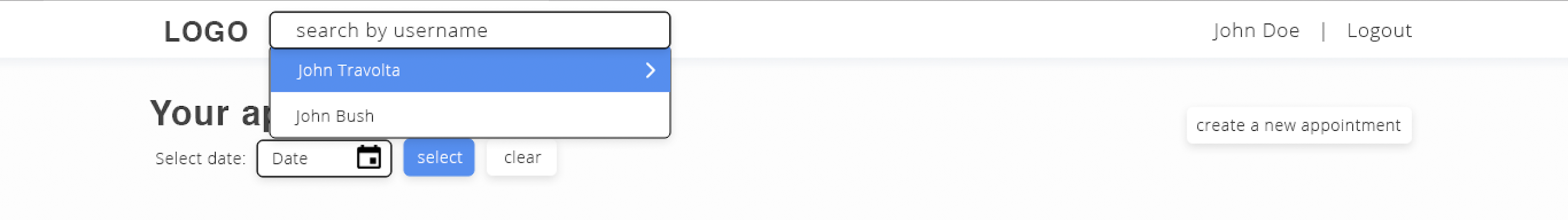


Fig. 6 Autocomplete

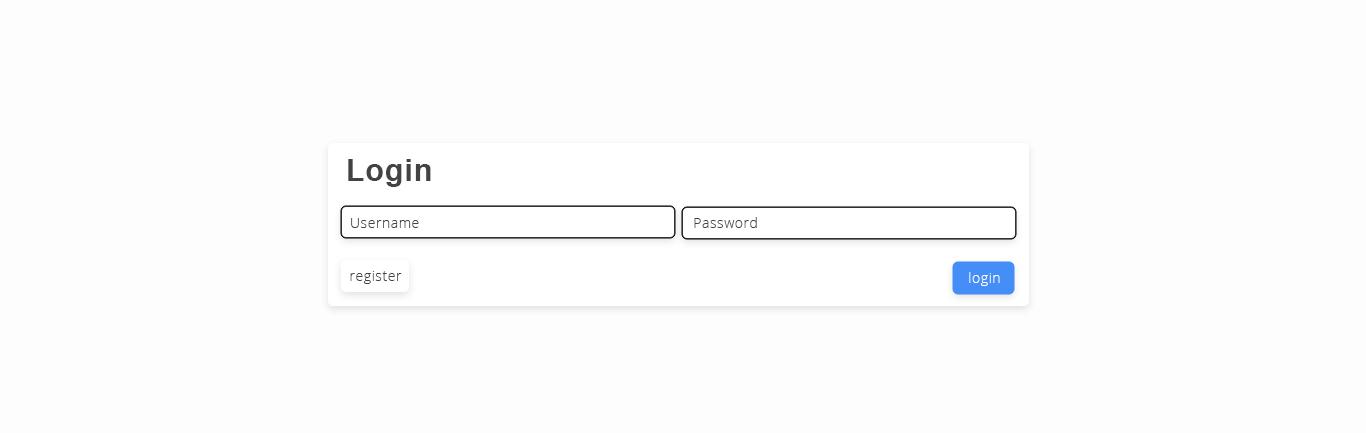


Fig. 7 Login page

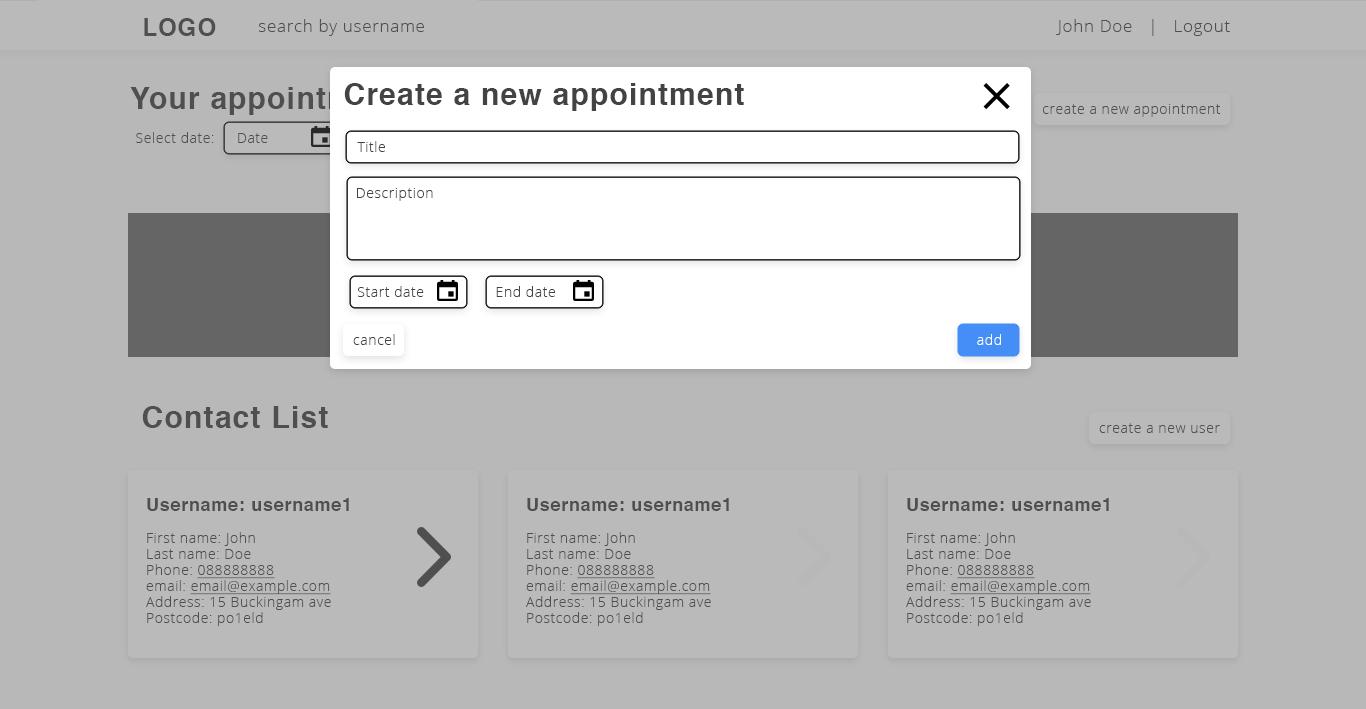


Fig. 8 New appointment popup

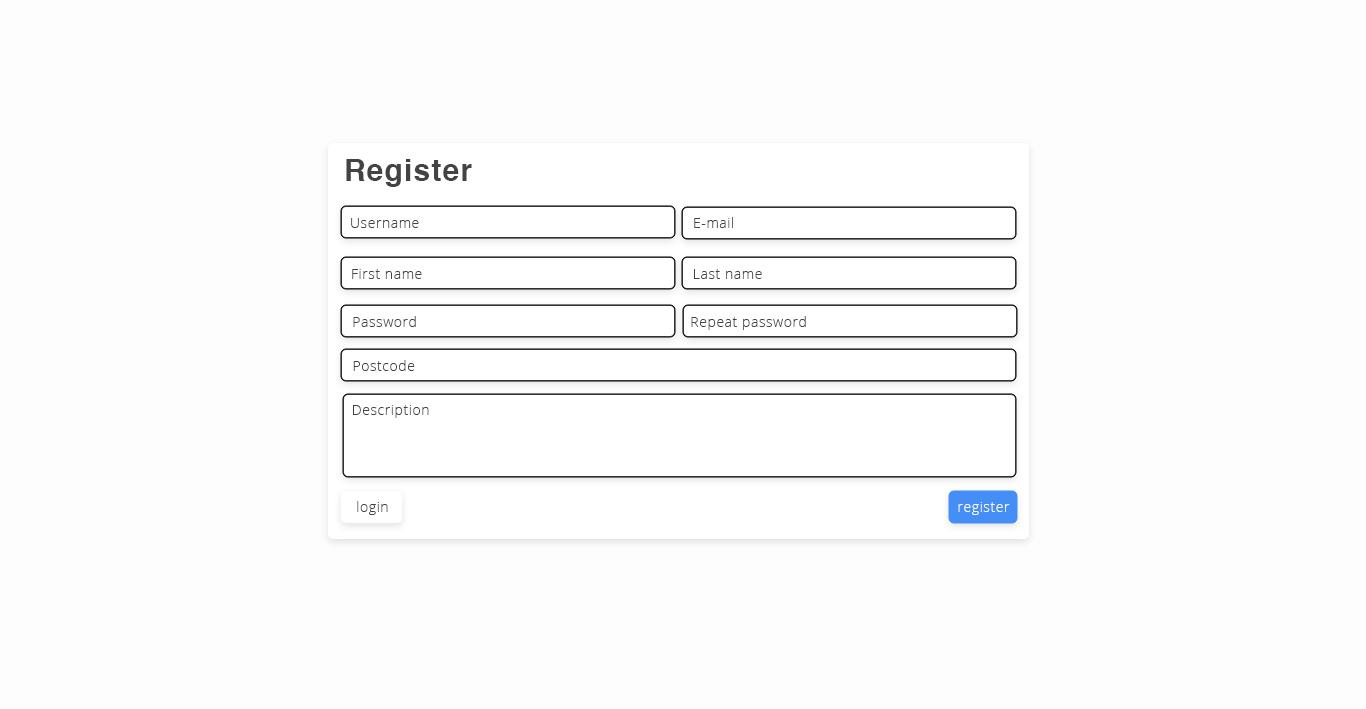


Fig. 9 Register page

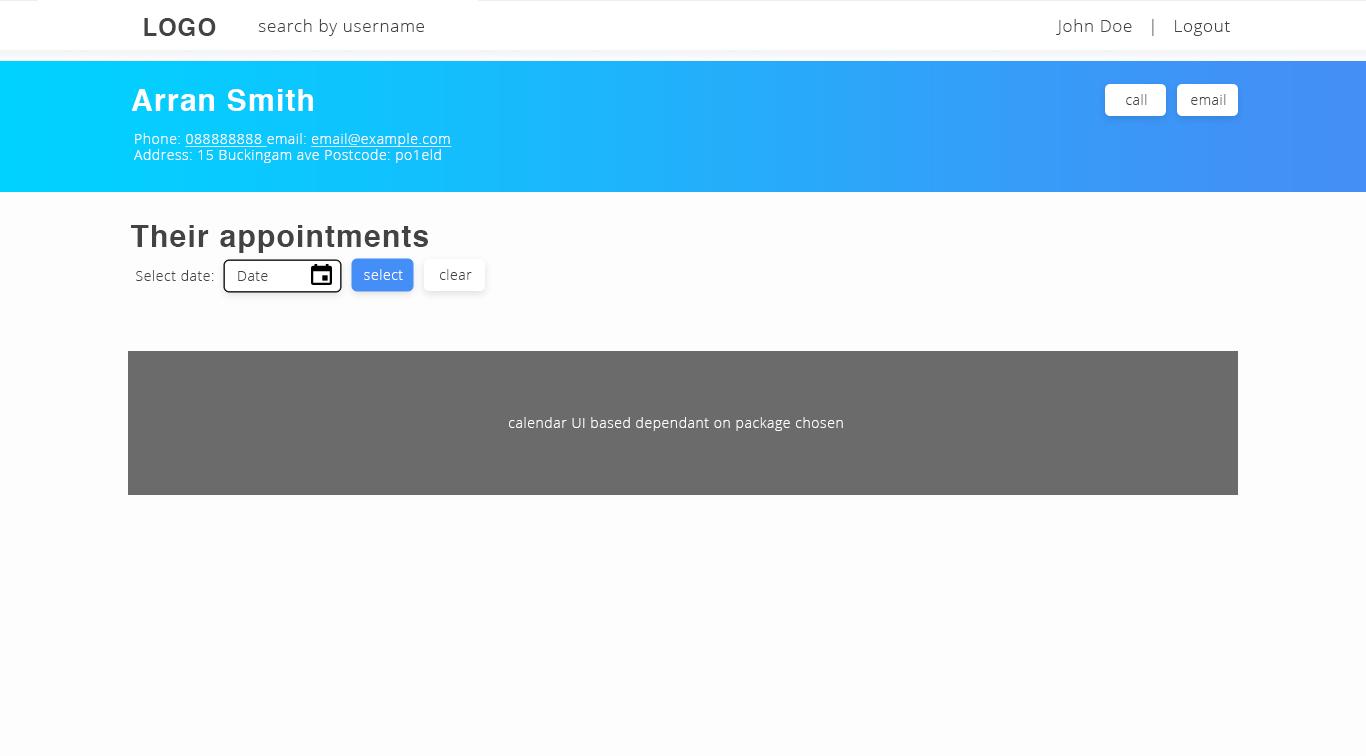


Fig. 10 Profile page

## 2.3 Flow of information

The system is only showing data to signed up users and thus the initial page is only pages they can access are login and register. After successfully registering, a user is asked to log in with the new user. After successfully logging in they are redirected to the homepage consisting of their appointments and all contacts. They can then visit the new appointment page, or the new contact page or view someone’s profile(see diagram at Fig. 11).

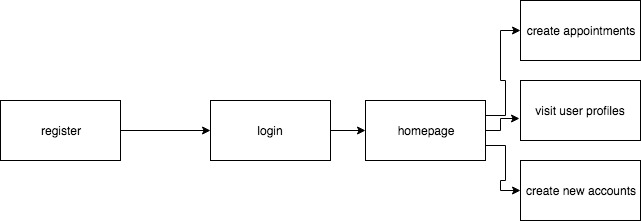


Fig. 11 Flow of information

# 3. Implementation and testing

## 3.1 Netbeans and Java EE

As an inexperienced Java developer, they were a lot of specifics to Java development which were not seamless to ease into. A trivial process like creating the database and creating the appropriate data pool took a substantial amount of time. It would have been very useful if there was a way for hot reloading components to speed-up development.

Java EE has a steep learning curve as well in terms of the project structure. Coming from php and javascript environments it was quite different.

## 3.2 Maven

Sometimes it was extremely tricky to debug errors returned from the server. Messages like “Transaction aborted” and lengthy logs didn’t prove to be helpful. From time to time a restart of the server or NetBeans would resolve an issue.

## 3.3 Database

Once an entity is changed(e.g added field) Maven will not create the field if it had already created the table. In this case, the table can be deleted and to build the project one more time. It was tricky to use JPA within the persistence layer especially for querying appointments based on users linked to it. Once again, the logging and error messaging were not helpful.

## 3.4 Time management

Poor time management resulted in an inability to finish all the features which were planned. Taking this experience into account, the author would try to use an iterative approach with strict milestones for future projects.

## 3.5 Testing

The artefact was tested with a variety of different invalid data entities manually. The author had intentions for unit testing however the timeframe and multitasking of many classed made that impossible.

# 4. Summary

The coursework did thoroughly improve the author's knowledge on Java EE development, complying with the MVC pattern, Object-oriented programming and XHTML. Even though the artefact was not finalized, the benefits were undisputable. In the future, a different server might be used - potentially spring. Things like hashing passwords should be explored in the future.

# 5. References

* Java Platform, Enterprise Edition: The Java EE Tutorial Release 7 - Contents. (n.d.). Retrieved April 30, 2019, from https://docs.oracle.com/javaee/7/tutorial/
* Goncalves, A. (2013). *Beginning Java EE 7*. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4302-4627-5