**Report based on Initial discussion at meeting**

**Topics discussed**

* Overall design, scalability and features of the system
* Market Analysis
* Technologies to be used
* Initial test plan

**The end goal of the system for the USER.**

*To bring an end to the estimated bill. Know exactly how much you are spending, and where you are spending it.*

**Design** - security driven development around maintainability and scalability

**Minimal System Requirements** - features of a minimal working system to begin development on.

1. Look through a list of their devices

2. Investigate their devices current and past status

1. Uptime

2. Metric (Temperature, Energy Used)

1. Must allow at least four devices to be monitored - but should allow more to be added.

2. Should accommodate growth to the UI and functionality.

3. Visual representation of trends.

4. Add/Remove devices

Note: More advanced solutions may utilise some interesting visualisation options

The core features of the system are:

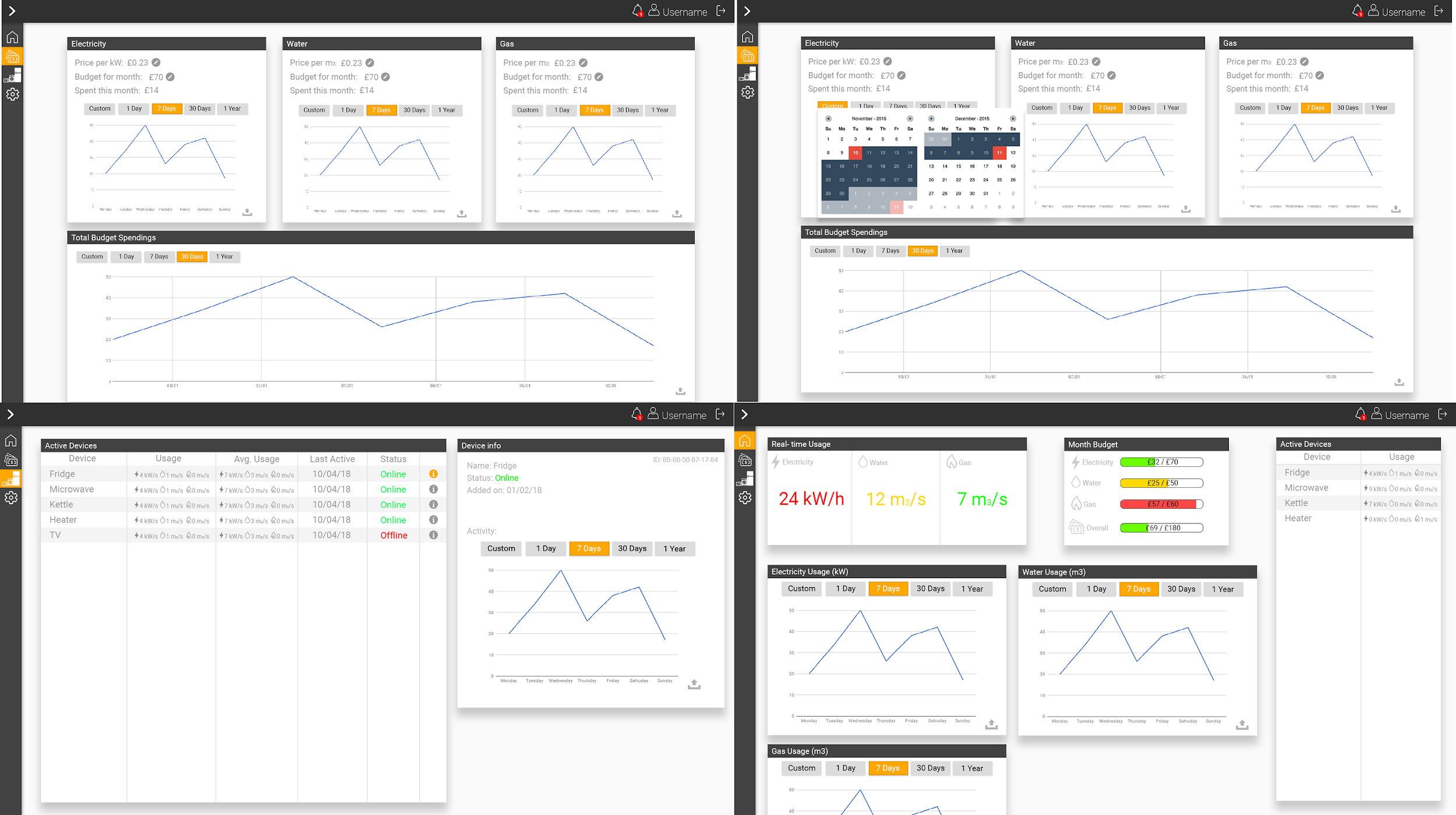
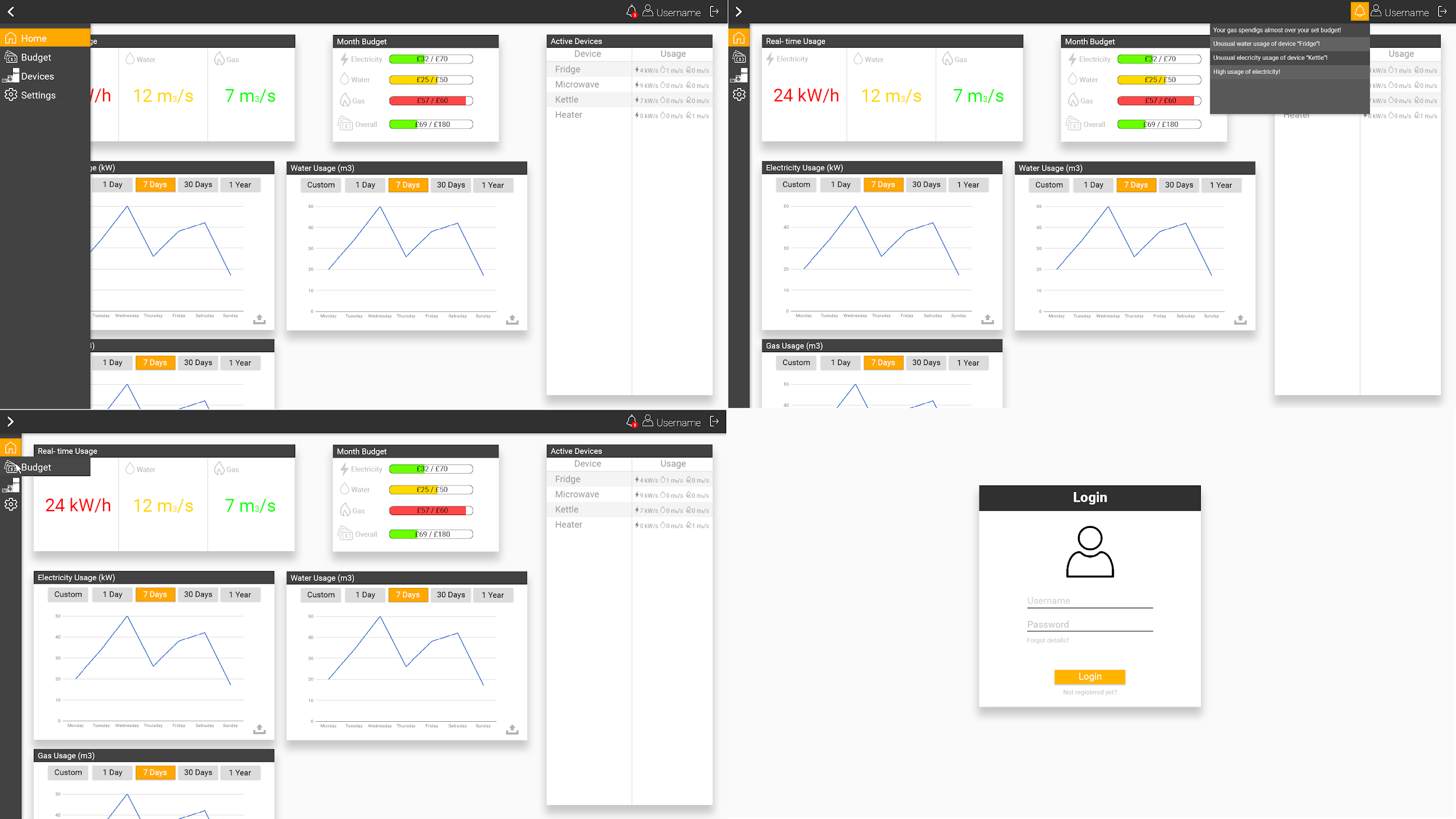
1. **Robustness** – the system should clearly show to the customer all data related to their usage of household utilities. Features that the customer will be able to see are:

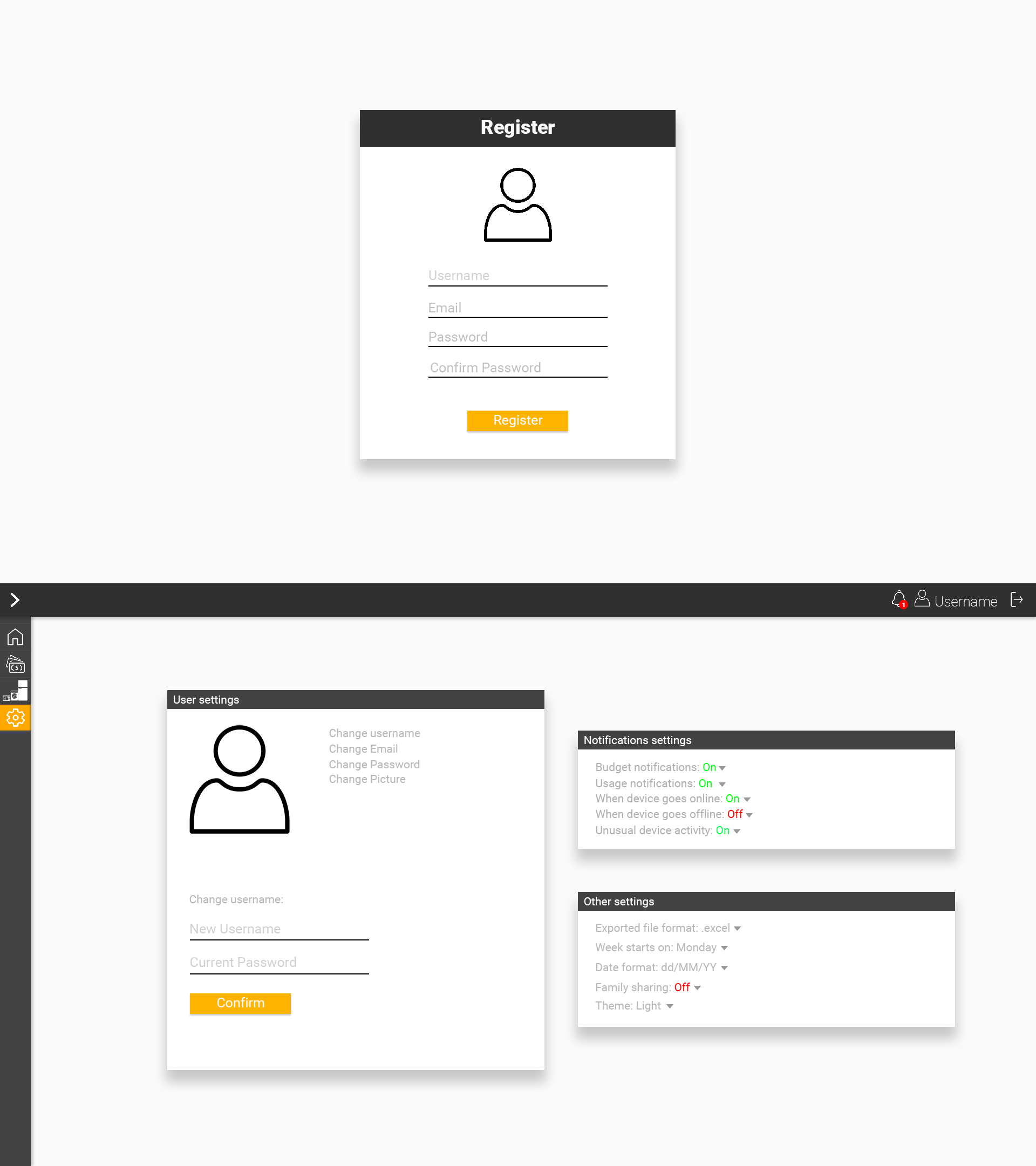
* Real time usage - The amount of energy that each device is expending in KW will be clearly displayed. A colour scheme will be used (Red: Using a lot more than usual, Yellow: using slightly more than usual, Green: standard amount).
* Real time graphs - As well as just seeing a current value for the Real time usage, the customer will be able to see this data displayed in a graphical format (kW y axis, time split per hour x axis), so that they can easily see how much they are using throughout the day. This will allow the user to spot trends within a day – is there a time where energy consumption is greater within a day?
* History of Usage– A history of the customers usage will also be displayed in a graphical format. There will be an option for the customer to switch the view of this graph between a day view, a weekly view, monthly view and yearly view. Are there certain days in a month where the customer is using more energy? Is there a particular week where energy consumption is greater, i.e. in the holidays? Does this change a lot in the winter months? Using these graphs will easily allow the customer to see this important data.
* Total current cost – Total cost in relation to a target cost set by the user. The user will be able to see how far away they till hitting their target total cost for the month. This figure will be displayed as a percentage as well as in a ‘bar’ that fills up as the current cost gets closer to the target.

*Suggestions for the user with basic AI. For example: ‘You appear to be spending a lot more than last week on heating? Reduce by 5% per day (0.01kW/h) to meet your monthly target.’*

* Reports – Reading graphs and analysing trends shouldn’t be left up to the user. An option to generate a report that will display this data in an easy to understand format will also be included. They could then simply save this PDF, print it off or have it emailed to them on a monthly basis.

1. **Scalability** – the user should easily be able to add new and remove new devices without experiencing a drop in performance. Adding a new device should hassle free user and will only require a click of a button and filling in some basic details (i.e. device name). The user will be able to start seeing real data usage of the device almost immediately (as long as setup was successful). Even if a user removes a device from their home, the usage history of that device will still be available for the customer to see.
2. **User Interface** –





1. **Security –** protection of user data is of high priority. Security driven development to help prevent SQL injection, CSRF (Cross-site request forgery) attacks and XSS (Cross-Site Scripting) attack vectors. Take the following measures to safeguard user data:

* **Encryption**
* **Backups**
* **OS / software updates / security patches on server.**
* **SSL**
* **Certain password strength required on user registrations.**
* **Optional 2FA**

**Technologies to be used**

CSS Framework: to be decided. Bootstrap 4.0 / MdBootstrap / Foundation etc

Javascript libraries for graphs:

* Chartist.js - <https://gionkunz.github.io/chartist-js/>
* Chart.js - <https://www.chartjs.org/>
* D3.js -

Any other suggestions.

Backend: PHP/MySQL. (Laravel?)

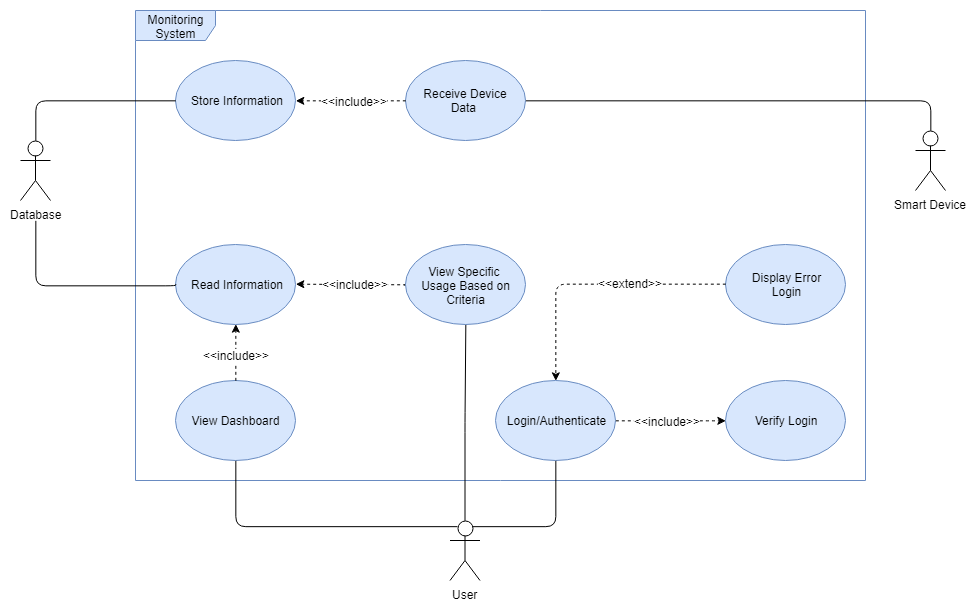
* Wouldn’t have to write own user authentication.
* Cleaner code with blade.
* Database maintenance more efficient with migrations.

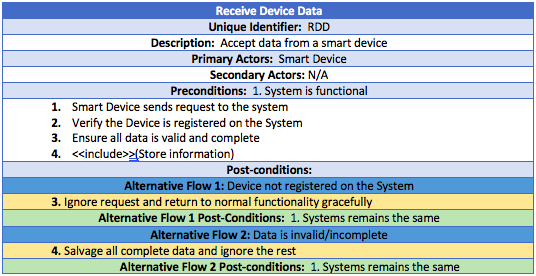
Version Control

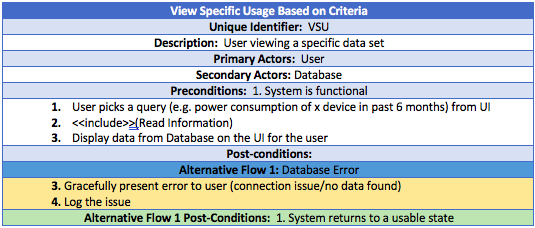
* Git
* Setup a Github team.

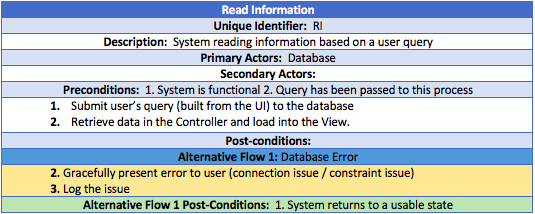
**Test Plan**

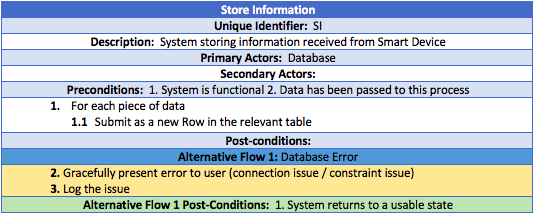
**Use Case Diagram**

****

****

****

****

****

**Application and Database Design**

(to be added)