Cloud Computing

Benjamin Cao
Jose Flores
David Jelley, Jr.

Goal

- Create an API to handle all requests.
- Create an expandable application framework.
- Create a secure database to store all data.
- Create a standardized user interface.
- Create management portals for developers, users, and Administrators.
- Create tools to help develop and test applications.
- Create a permission and user based system to allow private and shared access to data.
- Render and display the information provided by external devices as well as application input.

Motivations

- Data is sensitive, informative, and sometimes damaging.
- The complexity of the information contained needs to be conveyed accurately
- The relationships between seemingly unrelated data sets can be hard to detect.
- Data can be misinterpreted and displayed in ways that are deceiving.

Related Work

CareCloud

Primary:

- Stanford School of Medicine
- CareCloud

Secondary:

- Social media (Vast amounts of storage with “privacy”)
- Account-based websites (Google, GitHub, etc)

Mission

What is private?

Security standards are evolving as current methods become obsolete.

This group is investigating how data should be stored, transmitted, and shared while maintaining privacy and trust between the user and the data holders.

Features and Implementation

Database:

- Create well structured MySQL server and database

Web:

- A PHP API
- A PHP and Javascript framework that allows for independently callable widgets.
- Multiple access portals.
- Database and data management tools.
- Sand boxed Developer testing tools.
- Permission based user accounts

Database Mock-up

- Evaluate data produced by hardware.
- Evaluate need for additional tables.
- Add additional tables to current database design.

Current Mock-up

User Privileges

Doctor

- Able to view the records of all patients.
- Manage patients (Add, drop, update, transfer)

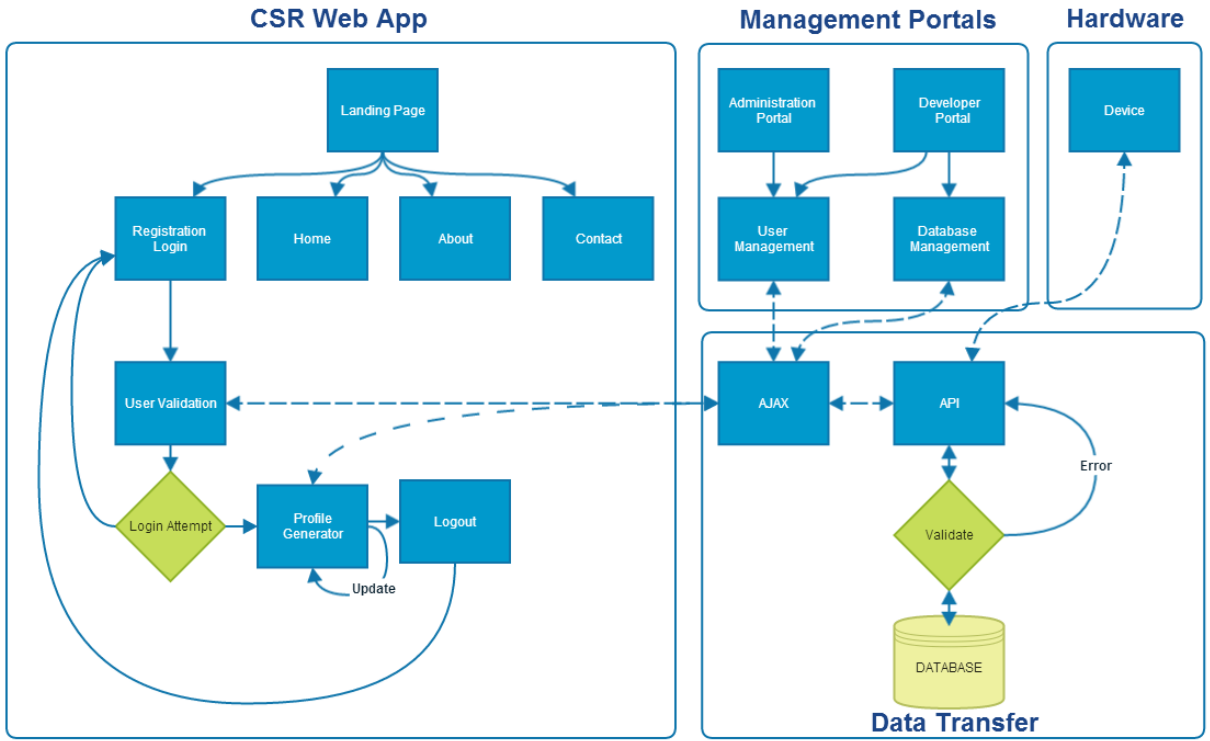
Patient

- Able to view their own records.
- Able to restrict view of data.
- Manage doctors (Drop, request)
- Manage observers (Add, drop)

Observer

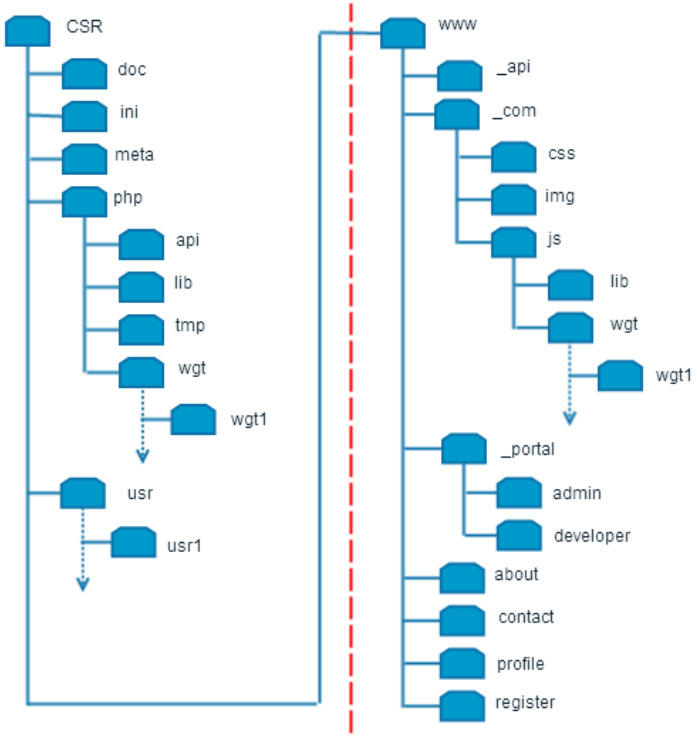
- Manage who they observe (Drop, request)

Approach - Application Flow



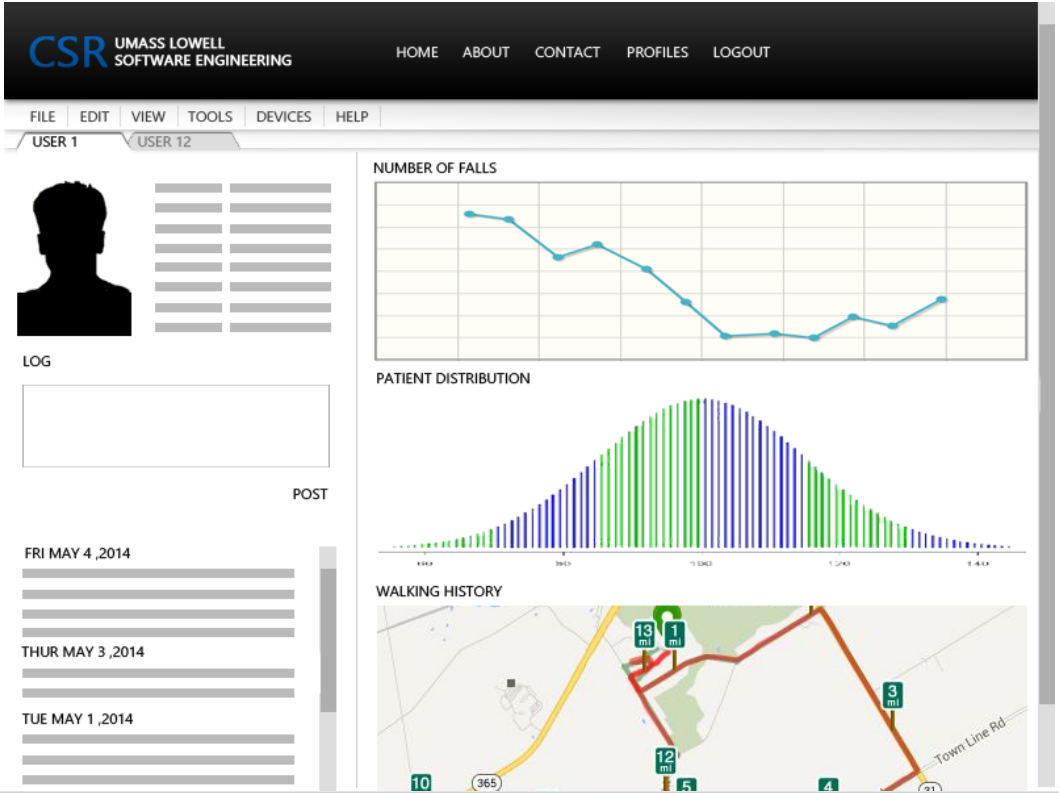
Application - Structure

Private
Server Side



Public
Client Side

Application - Interface



Demos

Live Demo

Developer Tool Demo

Evaluation

- Unit testing
- Component testing both separately and jointly. To validate and prevent application conflicts.
- Developer and peer focus groups.
- Privacy and security testing. We will validate account access and probe for weaknesses.

Timeline

Week	Goal
1	Proposal, presentation, and framework developed
2	AWS WAMP environment set up. Website template constructed. Developer tools constructed. Database finalized.
3 - 4	Create user validation scripts. Meet with device groups to discuss API needs. Create API. Testing and Bug fixes. Meet with device groups to discuss API capability.

Week	Goal
4 - 7	Research PHP/ jQuery UI and graphing widgets Create PHP web pages. Create application access portals. Write JavaScript and jQuery client side scripts. Testing and Bug fixes.
8	UI testing focus group.Testing and Bug fixes.
9	Finish styling and any animations
10	Testing and bug fixes
11	Final testing and submission.

References

Coding References

http://www.jquery.com/

http://www.jqueryui.com/

http://www.php.net/

Cloud

http://aws.amazon.com/
http://www.carecloud.com/

Security

http://med.stanford.edu/irt/security/cloud.html

Server

http://httpd.apache.org/
http://www.mysql.com/
http://www.phpmyadmin.net/