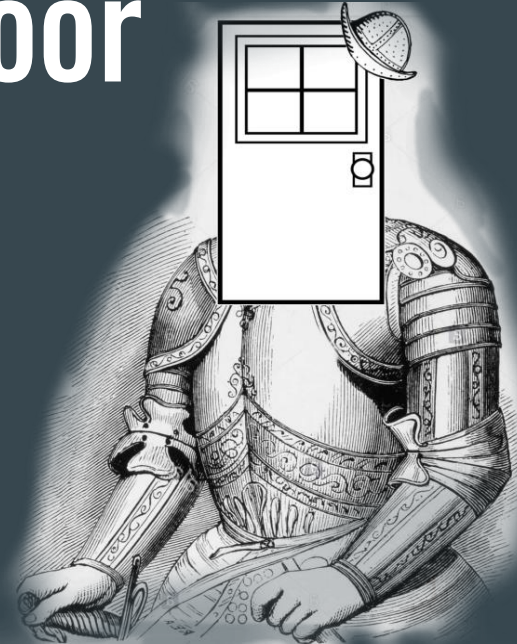
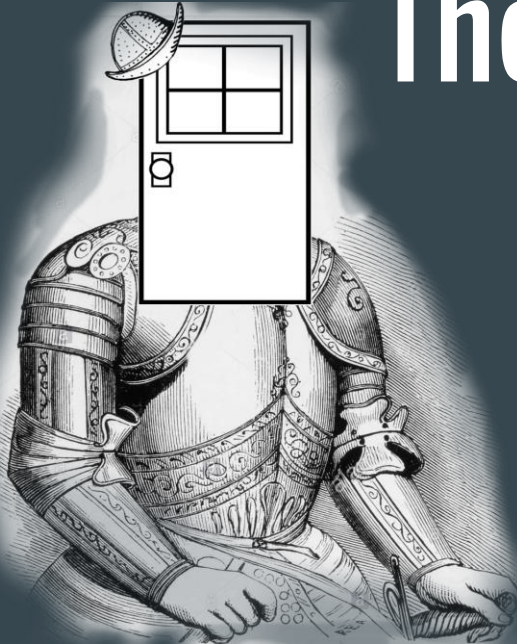


The Virtual Office Door

...

The Conquistadoors



Introduction

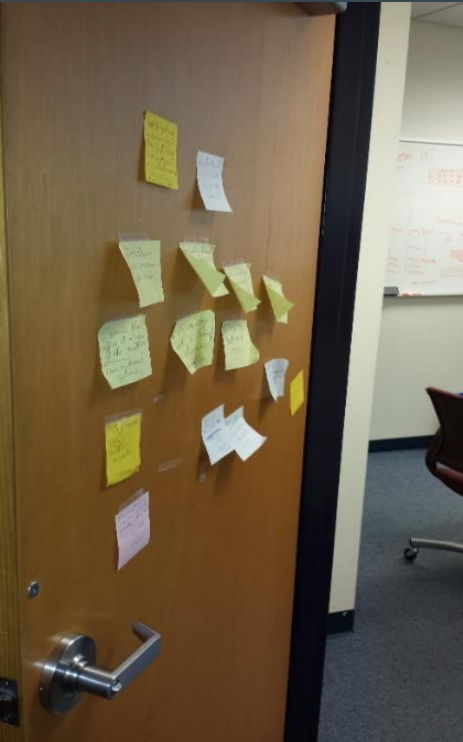
- Our Team: James Hauser, Mitchell Hewitt, Nicolas Melillo, David Snow ,Tyler Tollefson
- Our Mentor: Dr. Eck Doerry
- Our Clients: Dr. Eck Doerry and Michael Leverington

Problem Statement

- Office doors serve as a form of communication
- Many different people need access to office door information
 - Professors, Students, Managers, and working professionals
- Millions of people utilize office doors as a form of communication
- Communication is essential but physicality is a hurdle.



Problem Statement Cont.

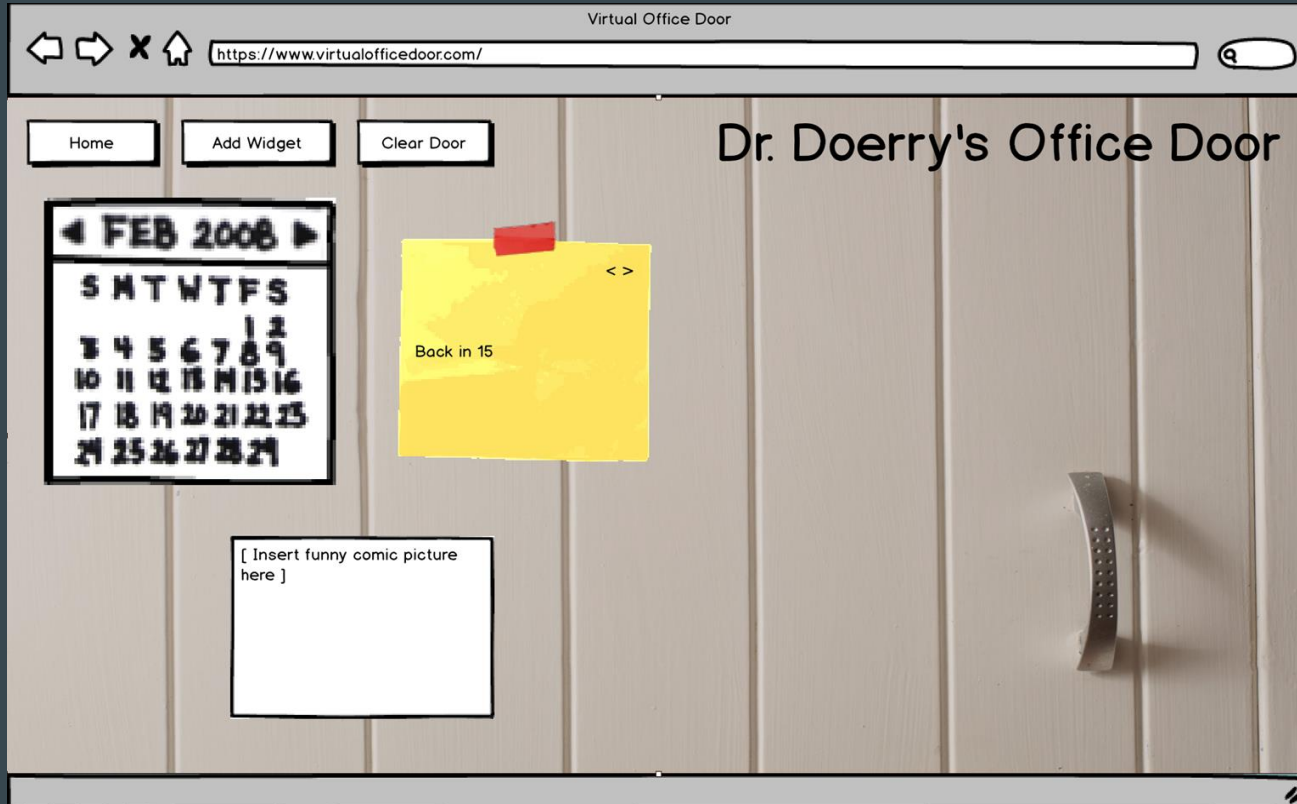


- Teachers are moving to the SICCS building on North Campus.
- Classes are still being held on South Campus though!
- Teachers still use office doors, but they become cluttered.
- Email communication with students is slow and unreliable
- Teacher office hours are always subject to last minute change
 - No way to communicate this to the students in a timely manner

Solution Overview

- We envision a secure, fast and account based Web 2.0 application that operates as a “virtual office door”.
- Their “office door” could display:
 - Calendar with availabilities and office hours.
 - Sticky notes for quick alerts.
 - And other useful “widgets”.
- Modular components that can be resized, repositioned and edited.
- Accessible on a laptop as well as a physical office door display.
- Allows users to communicate with door owners

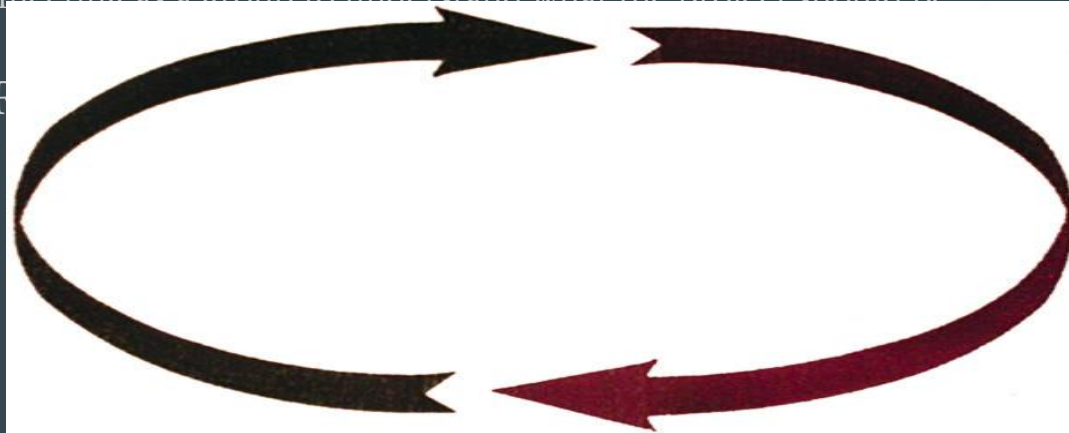
Solution Overview Cont.



Our Requirements Process

- Acquisition:
 - Meetings with our sponsors to get the broad User Domain Requirements
 - Meetings with the mentor to help refine those into Functional Requirements
 - Then meeting as a group to understand what the project should be

- Iterative R



Key User Domain Requirements

- UD 1: Website Login must be secure and reliable.
- UD 2: Users should be able to edit basic profile information
- UD 3: Users will be able to create a customizable “office door”
- UD 4: Support of multiple “apps” or “widgets” that will be utilized by the user
- UD 5: A touch screen display that will display the specific user’s “office door”
- UD 6: Use a cloud based backend architecture to store user and office door information

Functional Requirements

- 18 Functional Requirements
- For example:
 - “UD1.FR1. User must be able to securely sign up for an account using a google login system. Using the Google login API, user’s email will be registered with the backend and stored in the database. Using the backend application create a web space for the registered user.”
 - “UD6.FR1. Frontend will communicate with a database hosted on an Amazon cloud server. Use an architecture that allows quick and easy communication between the front end UI and the backend application. The back end application will need to handle get, pull, push, and delete requests.”

Nonfunctional Requirements

- Performance Requirements
 - Pull requests with the backend will take no more than 5 seconds to populate in the frontend UI.
 - It should take no more than 5 seconds, after user edits are completed, to update the database with current office door configuration.
- Environmental Requirements
 - The server must be ran on a cloud-based web service.
 - Display tablet must be affordable as well as easily configurable.

Feasibility Analysis

Google API

Integrate with our application

Amazon Web Service

Host website on the cloud

Django

Backend web framework

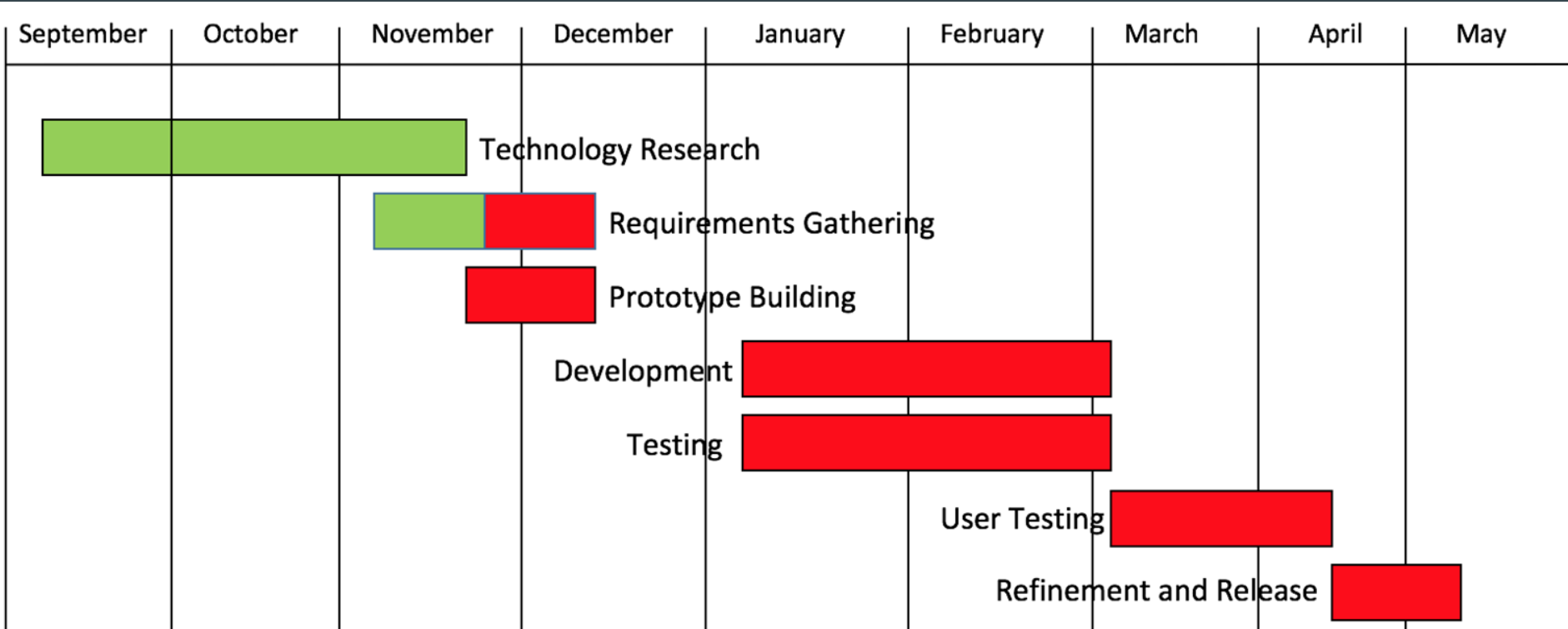
Python

UI with application

Risk Assessment

Risk	Risk Level (1 - Low and 5 - High)
Usage after introduction	2
Usability/Likeability by students	3
Application usage outside of Academia <ul style="list-style-type: none">• Copyrights	1
Amazon web service trial ending	5

Progress Update



Conclusion

- Communication is essential in the workplace.
- Time is money.
- Our Goal: To create a web application that allows for virtual office door communications between teachers and students, that could be expanded for use in different disciplines.

