

Office Door Kiosk

ECpE Senior Design Project

Team sdmay18-28

Project inspired and led by Dr. Thomas Daniels

Motivation

Problem Statement: Because professors are very busy, and it can be hard to get a face to face meeting with them even during their scheduled office hours.

Solution: The office door kiosk wants to provide a quick and convenient means of communication for students trying to make a face to face connections with their professor. The kiosk accomplishes this with features like requesting the professor to return, and by displaying relevant information to people who come by their office.

Users and Uses

Users:

- ◆ Iowa State Professors
- ◆ Iowa State Students

Student Uses:

- ◆ Send notification when at Kiosk
- ◆ Read notes posted by the professor
- ◆ Review the information provided by the professor.

Professor Uses:

- ◆ Receive notifications from students
- ◆ Post messages
- ◆ Remotely update any information on the kiosk from their phone
- ◆ Configure permissions
- ◆ Set office hours

Design Requirements

Functional Requirements:

- ◆ Remotely updated kiosk owner information
- ◆ Configurable Kiosk applications
- ◆ Leave notes for students
- ◆ Customizable calendar
- ◆ “Door bell” feature - alert professor that someone is at his door
- ◆ Quickly updateable office hours

Non-Functional Requirements:

- ◆ Must be tamperproof.
- ◆ Must have restricted hours.
- ◆ Must have responsive UI

Operating Environment:

- ◆ Outside of office doors
- ◆ Wherever a phone can be taken.

Engineering Constraints:

- ◆ Always has to be connected to the internet
- ◆ Will always need to be plugged in (Kiosk mode only)
- ◆ Has to be Device Agnostic

Standards:

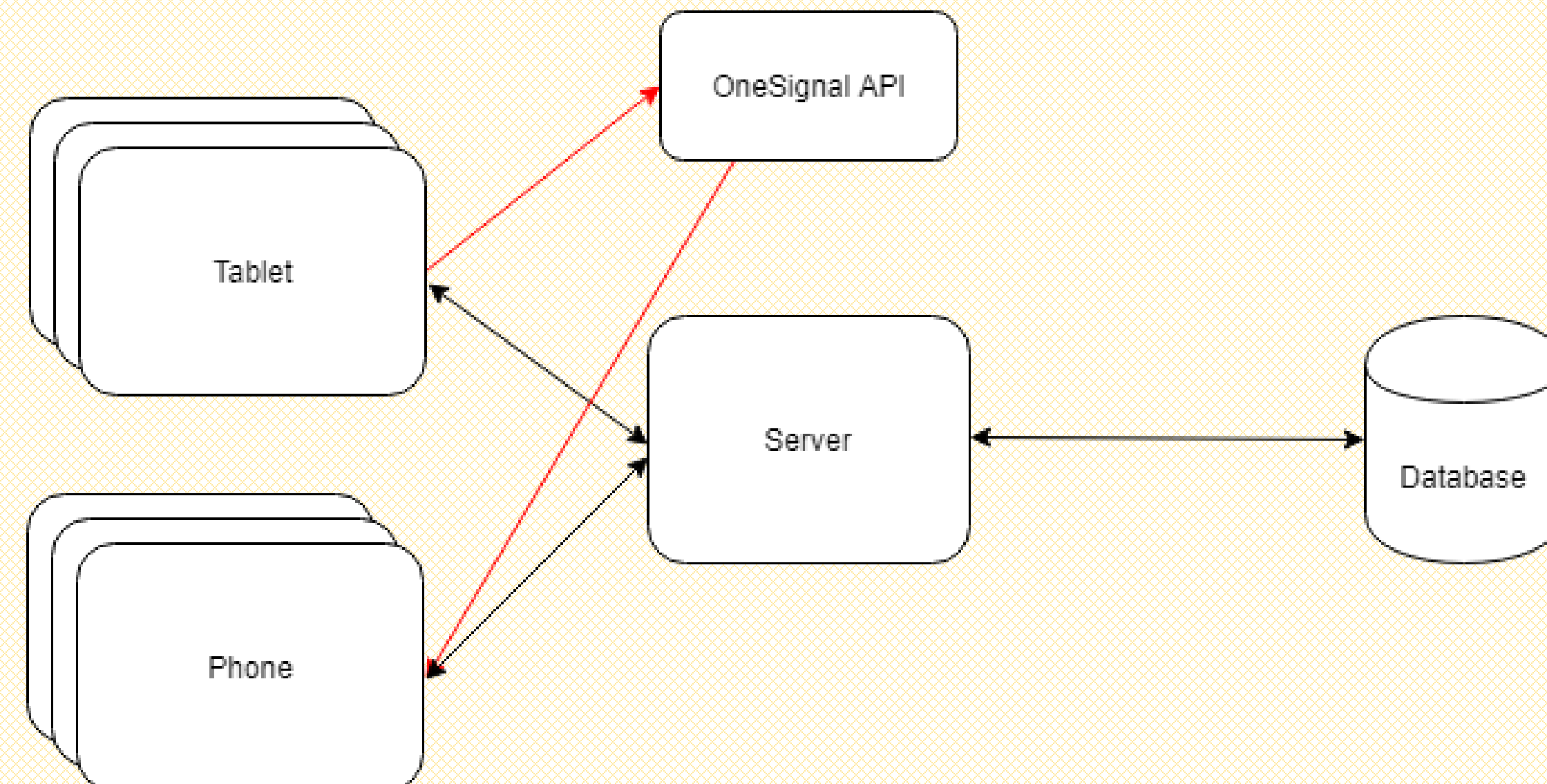
- ◆ ISO/IEC 21778:2017 - The JSON data interchange syntax
- ◆ ISO/IEC 16262:2011 -- ECMAScript language specification
- ◆ ISO 26000 - Ethical Standards
- ◆ IEEE 1028-2008 - IEEE Standard for Software Reviews and Audits

Design Approach

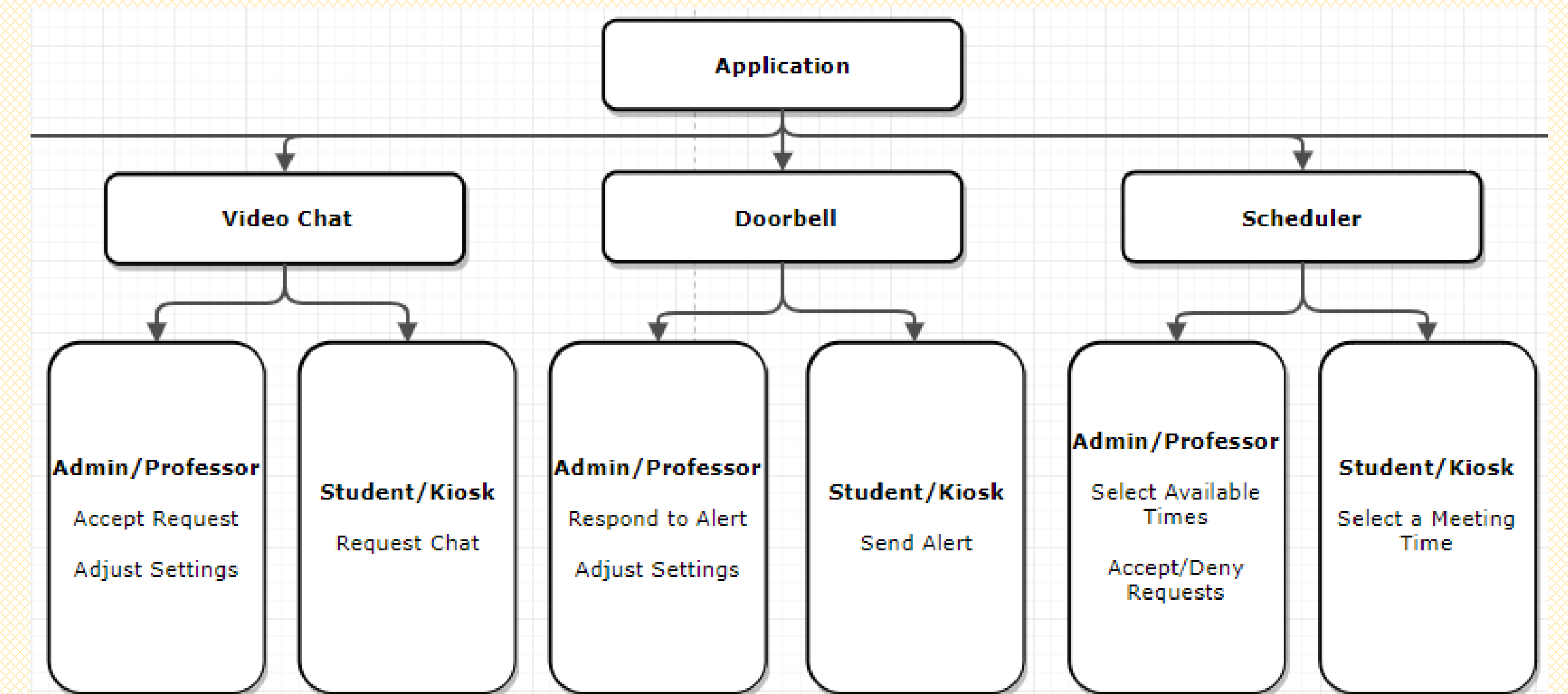
Concept Sketch



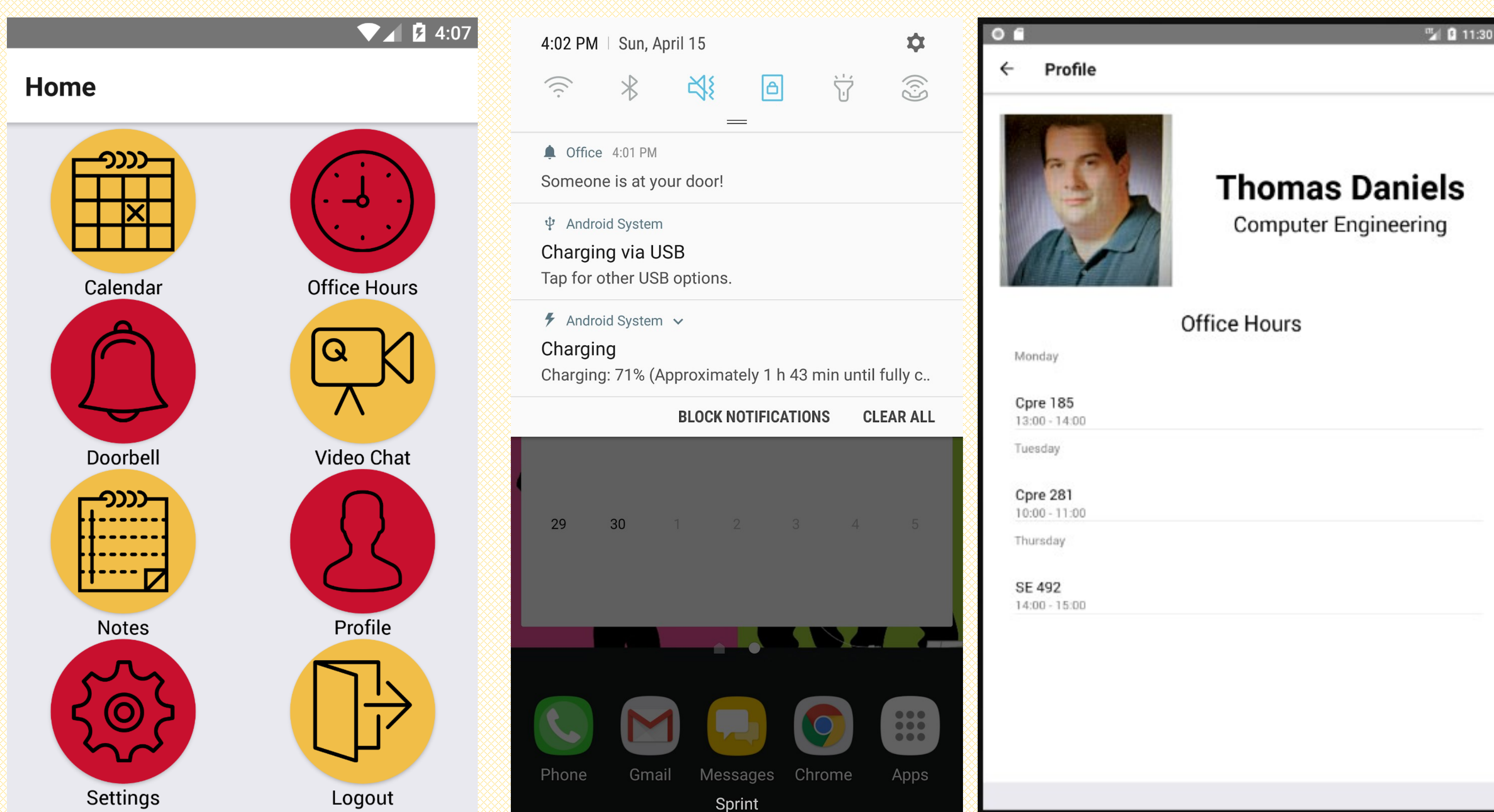
Block Diagram



Cross-section of Functional Decomposition



Resulting Application



Technical Details

Hardware used: (\$125.05 acquired through CDW)

- ◆ Amazon Kindle Fire 7
 - ◆ Peerless Universal Tablet Cradle PTM200
 - ◆ Tripp Lite Display TV LCD Monitor Wall Mount
- Software modules:**
- ◆ Languages: Javascript, HTML.
 - ◆ Libraries: React Redux, Shortid
 - ◆ Dev tools: Git, Visual Studio Code, Android Studio
 - ◆ Environments: Android, IOS
 - ◆ Frameworks: React-Native, Node.JS

Testing

- ◆ Emulators and physical devices to test use cases and user stories
- ◆ Jest: automated unit testing