Information needed:

* Pic of old clock system or the actual thing



**Purpose of Project/Initial Problem:**

GFS currently uses a time clock system for tracking time for hourly workers. This system allows hourly workers to track breaks and timestamp arrival and departure time. This system allows hourly workers to see their punch history, and punch in and punch out for work, lunch, and breaks using their ID Badges assigned to them by swipe/with the swipe of their ID badge and a press of a button that corresponds to that activity. A transaction or timestamp occurs when an employee swipes their ID Badge on the barcode scanner along the side of the device mounted on the wall and pressing a button that corresponds to the action they wish to record. A single time clock system costs $1800 and is provided by a third party/ outside source. Malfunctions or damage to the system is both costly and time consuming.

This project is multifaceted; one part hardware replacement recommendation and a second component software application creation. Currently GFS uses 130 wall mounted barcode scanning time clocks that are expensive to replace (roughly $1800 each). Our project is to develop an application which can serve to replace these aging time clocks and recommend tablet computing devices which can be easily deployed.

While this will be explored further in the feasibility study software considerations include building a web application using Angular 1 or building a native Android application. Either option will be run on tablet computers stationed within each GFS distribution center.

We are currently using dedicated purpose time clocks for tracking time for hourly workers.  We would like to replace these devices for several reasons:

* Save cost of devices (dedicated clocks are $1700-$1800 versus cost of tablet)
* Make repair/replacement easier (more vendors available)
* Enable use of clock replacements for other purposes
* Enable usage of personal phones to clock in/out from specific areas inside facility

1. Select mobile device to replace existing dedicated purpose clock.  Considerations include cost, durability, and ongoing maintenance/support.
2. Develop application that allows users to punch in/out.  Requirements include:
   1. Runs on any platforms, especially Android, IOS and Windows
   2. Has ability to run in connected & disconnected mode
   3. Has ability to identify user
      1. Minimally, must be able to scan badge
      2. Optimally, use biometrics
   4. Receives and stores basic information (employee, department) from central source
   5. Ease of use - minimal user touches to clock in
3. Propose alternate uses for mobile clocks.
4. Optional (as time permits): Enable ability for users to use personal phone to punch in/out.  Note that this must be restricted to certain areas within the building.  Ideas for implementing include geo-fencing and near field communication, but we are open to other ideas that meet these needs.

**Changes and Improvements:**

The entire clock system is being replaced by Lenova Tablet with an external barcode scanner. The tablet cost around $100 and the scanner is around $250 totaling $350 ultimately saving \_\_\_\_\_.

**Approach/Framework:**

* Native Android
* REST API
* External Barcode

Tooling Gordon Food Service makes use of Java and Angular 1 within their current software products. In order to achieve long-term supportability, we be using related tools in either the Java ecosystem, such as Android, should be pursue a native application, or Angular for a hybrid web-native application. Since we require hardware access, either to use a scanner or camera to process barcodes, a web application independent from hardware is not an option.

**Current Features :**

* Scan badge
* 7 button linked to actions: Clock In, Break In, Break Out, Lunch In, Lunch Out, Clock Out, Change Job
* Admin Functionality

**Prototype Application**

* Create wireframes
* Design application icon **Application Functionality**
* Application should be Multilingual, support Spanish, English and French Canadian
* Main screen will have one button "Scan your badge"
* "Scan your badge" button will activate Camera to scan employee badge barcode
* After successful scan, application will display 7 buttons, Clock In, Break In, Break Out, Lunch In, Lunch Out, Clock Out, Change Job
* "Change Job\* will open another screen to select job code
* After job code selection, application will go back to Main Screen
* Application should show who(Name of the person) clocked in
* Application should go to the Main Screen after any of the events
* Should not store any cache / If network is down, all punches must store locally
* Should work offline (Connected/Disconnected) / when Network comes up, application must push local stored punches to the server

**Validation**

* If employee clocked in for a day, "Clock IN" button will be disabled
* If employee has "Clock Out" record, All buttons will be disabled except "Clock in".
* If employee has "Break Out", all button will be disabled except "Break In"
* If employee has "Lunch Out", all button will be disabled except "Lunch In"

**Integration**

* Employee information push to the device
* Department information push to the device
* Clock data push from Android devices to Cybershift database via SOAP Service
* Local storage of Clock data - Disconnected mode

**Later features:**

Biometrics

Offline capabilities

**Title:** GFS Time Clock

**Collaborators:**

Kent Sinclair, Emily Peterson, Thanh Nguyen assisted by GFS Team