Logo

Description automatically generated

**K&O**

**Personnel Management Systems**

**User Stories**

**FOUO**

**User Story 1**

* **EPIC**
  + As an administrator, I want to be able to track my employees/student’s **attendance.**

**User Story 2**

* **Login Process**
  + The Login process will allow the user to create an account that will apply a randomly generated number. This number will be stored in a database with the user's Name, Position, Phone#, Email, Personalized PIN#, & potentially an image. After the user completes this process, a barcode will be generated with the user's unique ID# and emailed to the user. This would be used as a temporary means until an actual card could be issued.
* **Story Points: 1**

**User Story 3**

* **Store Account Data**
  + All the data that the user inputs after creating their account will be stored in a database. This database will be used to access data and verify login credentials.
* **Story Points: 1**

**User Story 4**

* **Create User Account**
  + As a user, I want to be able to create an account, so I can access the user portal to clock-in and out.
* **Story Points: 2**

**User Story 5**

* **User Login Screen**
  + As a user, I want to be able view a login screen, so I can scan my ID card and enter my pin code.
* **Story Points: 1**

**User Story 6**

* **Retrieve Data(Remote Setup)**
  + Program should iterate through the database and ensure that the user has inputted matching UserID & PIN#.
* **Story Points: 3.5**

**User Story 7**

* **Store Time Data**
  + **After user has created an account & logged in successfully, the current time and date will be stored in a database tracking each user's activity.**
* **Story Points: 1.5**

**User Story 8**

* **Display Output**
  + **Must meet current requirements but can be used to make visual improvements throughout the process.**
* **Story Points: 2**

**User Story 9**

* **Front-End User Login Screen**
  + **Coding for JSP & HTML**
* **Story Points: 4**

**User Story 10**

* **Front-End Create User Account**
  + **Coding for JSP & HTML**
* **Story Points:5**

**User Story 11**

* **Back-End Create User Account**
  + **Coding for JSP & HTML**
* **Story Points: 3**

**User Story 12**

* **Back-End User Login Screen**
  + **Coding for JSP & HTML**
* **Story Points: 3**

**User Story 13**

* **View Admin Data**
  + As an admin user, I want to be able to view an admin table that displays current Administrator data, so I know who the current administrator in the organization.
* **Story Points: 2.5**

**User Story 14**

* **View Employee Data**

As an admin user, I want to be able to view an employee table that displays current employee data, so I know who works for or is a part of the organization.

* **Story Points: 1.5**

**User Story 15**

* **Registration Notification**

As a user, I want to be able to view a webpage that displays "Registration Accepted", so that I know my registration form was submitted successfully and requires review.

* **Story Points: 6.5**

**User Story 16**

* **Registration Error**

As a user, I want to be able to view an error webpage(404), so I know that I ran into an issue with the registration process.

* **Story Points: 1**

**User Story 17**

* **Save Picture Links for the Types of Users**

When pictures are taken, users in do should create a link and save it into database.

* **Story Points: 1**

**User Story 18**

* **View Work Table**

As an admin user, I want to be able to view a table with current employees on shift, so I know who is currently logged into the system.

* **Story Points: 1**

**User Story 19**

* **Unique Barcode Display**

This issue is used in the creation of a unique barcode after the user creates a new account.

* **Story Points: 1**

**User Story 20**

* **Print User Results**

Email the user their information that they just created. That way they have the barcode and pin to scan into work/school.

* **Story Points: 1**

**User Story 21**

* **Archived**

**User Story 22**

* **Archived**

**User Story 23**

* **Connecting to Database**

Create a class to allow the program to read from a csv file with the information necessary to connect to the database.

* **Story Points: 4**

**User Story 24**

* **Responsive Webpage**

As a user, I want a responsive website, so I can view the website on various devices(laptop, desktop, mobile devices).

* **Story Points: 4**

**User Story 25**

* **Secure Website**

As a user, I want to be able to login into a secure server, so I know that my data is safe.

* **Story Points: 6.5**

**User Story 26**

* **Content Security Policy**

Help detect and mitigate attacks which include data injection and cross site scripting(XSS). This is an additional layer of security to protect sensitive information.

* **Label:** Enhancement
* **Story Points: 4**

**User Story 27**

* **Successful Login/Logout**

Create pages to show the user that their inputs were successful.

* **Story Points: 4**

**User Story 28**

* **Lock User Accounts**

As an administrator I want to lock user accounts after 3 consecutive incorrect PIN number entries. Resulting in the user being required to contact administrator to unlock the account for continued access.

* **Story Points: 2**

**User Story 29**

* **Secure Hash Algorithm**

As a user, I want my sensitive information to be secured by use of SHA, so I know my information is protected by potential hackers.

* **Story Points: 2**

**User Story 30**

* **Conceal SQL Statements(SQL Injection)**

Hide SQL Statements within the program for added security to the webapp.

* **Story Points: 2**

**User Story 31**

* **Main System Login/Logout**

As a master admin, I want to be able login and logout of the main system, to limit user/admin power IAW defense-in-depth strategy.

* **Story Points: 1**

**User Story 32**

* **Incorrect Input Page**

Create a webpage to show when the user incorrectly enters credentials.

* **Story Points: 2**

**User Story 33**

* **Locked Account Page**

Create a webpage to show when the user's account is locked and must contact an administrator.

* **Story Points: 2**

**User Story 34**

* **Reset Attempts**

Reset user's wrong credential attempts after successful try prior to the 3rd time.

* **Story Points: 2**

**User Story 35**

* **Error Redirects**

Trying to crash the program and find holes that we missed while creating the program.  
Created new error pages and fixed some bugs in the backend for the sql statements.

* **Story Points: 4**

**User Story 36**

* **Reset Locked Accounts**

Give the admin the ability to unlock a locked account and reset the user's pin.

* **Story Points: 4**

**User Story 37**

* **Update Admin Information**

As an administrator of the program, I want to be able to edit my information.

* **Story Points: 1.5**

**User Story 38**

* **Admin Task List**

As an admin user, I want to be able to add and delete task, so I can better track my upcoming appointments.

* **Story Points: 4**

**User Story 39**

* **Automatic Emailing**

As a user and administrator, I want to receive an email stating that I have successfully created an account with the information previously submitted.

* **Story Points: 13**

**User Story 40**

* **Admin Clock-In**

As an admin, I want to clock in immediately after logging into the home page.

* **Story Points: 2**

**Notes**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

[**RETURN**](#_1_Introduction)

**This Page Left Blank Intentionally**