Tanner Reits

Steven Hawes

Audre Staffen

Edgar Cardenas

Eagle Technologies Time Management CORE

Description

We will be creating a time management application for Eagle Technologies Group. This will be a web application hosted initially on their local servers and then possibly migrated to an azure cloud environment. It will integrate Eagle’s on-premise SQL database for data acquisition and storage. We will also be creating an SSIS job to run nightly on the temp-tables we will be using for user input values.

Standards

All back-end code will be written in C# using the ASP.NET framework. It will also integrate with Microsoft SQL Server for database communication. All our back-end code will be written to follow the .NET standard. Front-end code will be written in HTML and CSS with Bootstrap for mobile compatibility. The front-end development will follow Google’s HTML standards.

Database

Eagle currently uses Microsoft SQL Server 2014 for hosting their employee records database. We will be creating temp-tables within the database for storing the user input time entries before they are written to the live database each night.

Version Control

We will be using Git as our version control standard. Git is an easy to learn and use system that integrates well with most operating systems and programming languages. Git repositories also allow us to store other documentation and files along with source code. In addition, Git integrates natively with Microsoft Visual Studio making version control and branching effortless.

Maintainability

If the application stays hosted on their local servers, Eagle stated they will bring in developers if necessary for future maintenance as they are not native developers. Regardless, they will be responsible for future maintenance and development.

Security

Microsoft has a webpage on security in ASP.NET applications for its MVC frameworks. We will follow their documentation for user authentication/authorization and securely storing information within the application configuration file.

We will also use information found on the Ruby on Rails security page of common issues and translate them to ASP.NET to ensure our application is secure from both internal and external threats.

Testing

We will be conducting unit tests, system tests, usability tests, and black-box tests. A security scan will also be necessary as this application will be accessible from outside of the company network. The testing document goes into detail about which tests will be performed and the tools that will be utilized.

Use Cases

1. User logs in to application using Total ETO login credentials as verified by the on-premise database. Database password must be decrypted and verified against user input. Both username and password must match, error is thrown if no match is found.
2. After login, user is redirected to the “home” screen. This displays their current time entries if they have any as well as allows them to add a new entry. When adding a new entry, employees must enter the date, project number, job code, hours worked, and whether they were on the road. All available data will automatically be populated in dropdown menus using stored procedures. If all entries are not entered, an error message flags the unentered fields. Upon successful entry, the new data will be displayed at the top of the page in the current entries.
3. Users will be able to navigate to the edit page to edit any of their current, un-approved entries. When a user selects and entry to edit, it will populate all the fields with the select entries data and the employee will be able to change them to the correct values. Users may submit their new entry and the temp-table will be updated, or the employee can delete the selected entry and it will be removed from the temp-table.
4. If a user has manager status (as determined from the managers table) they will be able to access current time entries for all of their respective employees. Managers will be able to add, edit, delete, and approve any open entry for an assigned employee.
5. Each night, an SSIS job will push all entries from the temp-table to the live database table. It will then generate a report of entries for each active company job to get a total number of hours spent on the project so far as well as break-downs of each job on the project.

Future Releases

1. Migrate time entry application to an azure cloud environment using developer-written API’s for communication with on-premise database.
2. Implement RFID integration. Users on site will be able to use their company RFID badges to quickly log into the application using designated workstations.