eMoney Advisor

Application Deployment Tracking System

ELECOMP Capstone Design Project 2018-2019

Sponsoring Company:

eMoney Advisor LLC.
100 Westminster St.
Providence, RI 02891
http://www.emoneyadvisor.com

Company Overview:

eMoney Advisor is a leading software development company in the FinTech industry. We develop cutting edge web applications used by thousands of financial advisors to plan their clients' financial success. There are over 25 Agile Scrum teams working from locations in several locations around the world to bring these excellent tools to market.

We believe in the Agile principles of self-directed and empowered teams. Staying current in the technology and features of our application is a challenge we gladly accept. We love digging into new technologies and tools just to see what is possible and what sweet capabilities we can add to our product. We are looking forward to partnering with talented engineers from URI to make eachother better.

See our website for more details: https://emoneyadvisor.com/careers/

Technical Directors:

Daniel Jaquez
Senior Software Engineer
djaquez@emoneyadvisor.com
https://www.linkedin.com/in/daniel-jaquez-48666123/

Gary Jutras
Senior Software Engineer
gjutras@emoneyadvisor.com
https://www.linkedin.com/in/garyjutras/

Project Motivation:

We are continuously adding new features and making our web applications better. To get these innovations in the hands of our customers quicker we need a system that will track the status of our application during its release cycle. During a release there are several steps that require the input of many team members. No system exists that is able to track the status of each of these steps and display them. [1] This creates some issues in the release process. First, when issues arise during a release it is not easy to communicate what has gone wrong and what needs to be done to solve the issues. Second, there are times when the release is held up waiting for tasks to be completed.

Delays in releasing have an impact on the ability of a company to react. Reaction time, given the right circumstances, can be costly. The application we propose to build would help to reduce the release cycle time of our application and therefore reduce reaction time of our company.

Anticipated Best Outcome:

We expect to have an operation system with a core set of features that is capable of tracking the status of a release. The application will be manually updated through user interaction. It will display the status of the current release and past releases. When a person has not completed a task in a configurable amount of time the system will send a reminder about the task.

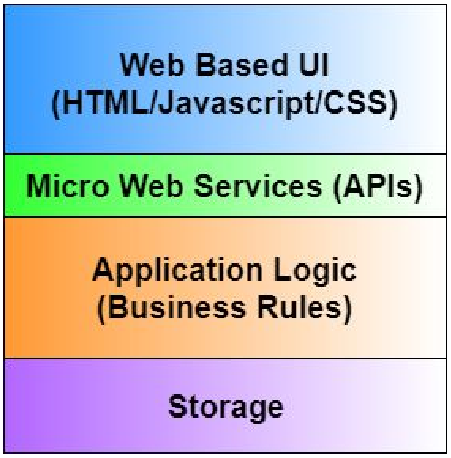
[1] We have looked into several applications that do not have the specific set of features we are looking for. These applications work well for their intended purpose. Some track the actual deployment of the release (Octopus, Electric Cloud). Others track work through the development cycle (Jira, Pivotal Tracker). GitLab even combines the tracking of work through the entire development cycle. None of them track the release process itself through the various steps to production.

Project Details:

Overall System Concept:
The system will have a concept of a release. This release will move through a release process. The release process is made up of steps. Each step will need to be completed before the release moves on to the next step. To complete a step the actions of that step will need to be taken. If the actions are not taken in a configured amount of time the system will send notifications to people assigned to the task as a reminder.

There will also be a dashboard that will show the state of the most recent release. The dashboard will show what the current step is in the release and who is responsible for that step.

Technology Stack:
This section defines the expected technology stack. The technologies that are specified are expected to be used. Any suggested technologies or technologies not specified that are needed to complete the project will be specified by the team and will need reasoning behind the selection made.



The application will consist of several layers.

1. Web Based UI: The application user interface will run in browsers both on mobile devices and on desktop devices. The suggested technology will be to use React.js with Redux for data flow.

2. Micro Web Services: The web UI will connect to RESTful APIs that will provide the data services needed to handle all aspects of the system. The technology will be written in ASP.net Core web APIs.
3. Application Logic: This is the central codebase that will be shared across any platform (mobile app, web app, etc.). It contains the business rules that make the application work. It will be implemented in the latest version of .Net Core.
4. Storage: This is the data storage layer where the state of the application will be saved. The suggested technology for this layer will be a modern version of SQL Server.

An investigation into using Microsoft Azure web services will be completed and a recommendation from the team will be made to support the selection of self-hosting vs a cloud hosted solution.

The design of the application's user interface and code structure will be developed over the life of the project.

Composition of Team:

2 Computer Engineers

Useful Skills:

The following list is a set of skills that will be used in the project.

Useful Computer Engineering Skills:

- Object-oriented Programming
  - Language (C#, Java, C++, etc.)
- Web Application Development
  - HTML
  - Javascript
  - CSS
- Databases
  - SQL

In addition to these skills there will be a focus on modern software development principles and processes. Some key ideas to be familiar with are Test Driven Development (TDD), Micro Services, and SOLID software design principles:

S – Single Responsibility: An object should only have one purpose.
O – Open/Closed: Software is open to extension but closed to modification.
L – Liskov Substitution: Objects of the same type should be able to be swapped without breaking the application.
I – Interface Segregation: Clients should not be forced to depend upon interfaces that they do not use.
D – Dependency Inversion: High-level modules, which provide complex logic, should be easily reusable and unaffected by changes in low-level modules, which provide utility features.

The most important attribute of a participant of this project is the willingness to learn. If you love working on tech and figuring out how to make something work in code then you will do well with us.

Anticipated Best Outcome’s Impact on Company’s Business, and Economic Impact

The primary purpose of this product is to reduce the release cycle times. The ability to release new features in a timely manner improves a business' reputation with their customers by both reacting to their needs and presenting exciting new possibilities. Developers of Facebook are told to “move fast with stable infrastructure” - favoring new functionality in place of perfection. Additionally, a faster release cycle will enable eMoney to test more features with their customers; getting critical feedback to improve their products even further. This accelerated release timeline will increase eMoney's competitive offering; improving customer satisfaction and resulting in greater total revenue.