

The Motivation Bot:

Andy Nguyen, anguyen4@stevens.edu

Erik Buczek, ebuczek@stevens.edu

Dominic Ortiz, dortiz@stevens.edu

Written: March 29th, 2020



I pledge my honor that I have abided by the Stevens Honor System

Signed:

Erik Buczek

Date: 3/29/20

Andy Nguyen

Date: 3/29/20

Dominic Ortiz

Date: 3/29/20

<https://github.com/ebuczek525/SSW-345-Bot-Project>

Problem Statement:

Motivation Bot will try to make workers more productive by both encouraging and rewarding productivity through a communication-based worker community. Although many companies make use of communication applications such as Slack, many do not utilize them to their full potential. Worker productivity can be a very detrimental issue when it is at low levels, as an unproductive workplace can lead to a loss in revenue and growth. Companies and projects want workers who are motivated and work efficiently, however, many workers feel disinterested in working and can get distracted. Workers can be productive and motivated by being rewarded through gamifying their experience while helping growth and progress on projects and in companies.

Bot Description:

The motivation bot will be implemented by combining some of the aspects of the RewardBot and FocusBot. The bot will exist on a worker communication tool, such as Slack, while being linked to a project on a Kanban/Scrum board. Thus, the product functionality will be twofold. The bot will provide productivity aids, while also providing rewards once the desired productivity metric has been accomplished. For example, a worker will receive a point-based reward for completing a task on their Jira board that was reminded of them by the Slack-based bot. The bot will need to maintain a memory and knowledge of all the users as well as tasks that have been assigned to each user, which means it will incorporate responders in order to handle the data and trigger responses.

We believe that if our bot is incorporated into everyday business processes, employees will be more aware of their work, motivated to finish it, and proud of the goals they achieve.

Get Motivated, Get Productive, Get Money

2 Use Cases:

Use Case: Set Productivity Goal

1. Preconditions

None.

2. Main Flow

Users provide a point value productivity goal that they wish to set for the day/week/month[S1]. Bot will display the average point value for all the tasks, and a list of all tasks[S2]. Bot will set a suggested pace in which the user needs to complete tasks[S3].

3 Subflows

[S1] User chooses either a daily/weekly/or monthly goal and sets an integer goal.

[S2] Bot will return a list of open tasks. Bot will calculate the average task point value.

[S3] Bot will set and display the suggested pace to complete this goal.

4 Alternative Flows

[E1] No open tasks

Use Case: Task Reminder

1. Preconditions

1 or more tasks taken

2. Main Flow

Users take on a task and expected finish date/time[S1]. User sets a reminder frequency[S2]. Bot will send a reminder message at determined frequency until silenced, will send late messages if expected date/time is not reset[S3].

3 Subflows

[S1] User takes a task and sets a due date in MM:DD:YYYY HH:MM format

[S2] User inputs reminder frequency in Hours or Days as Integers

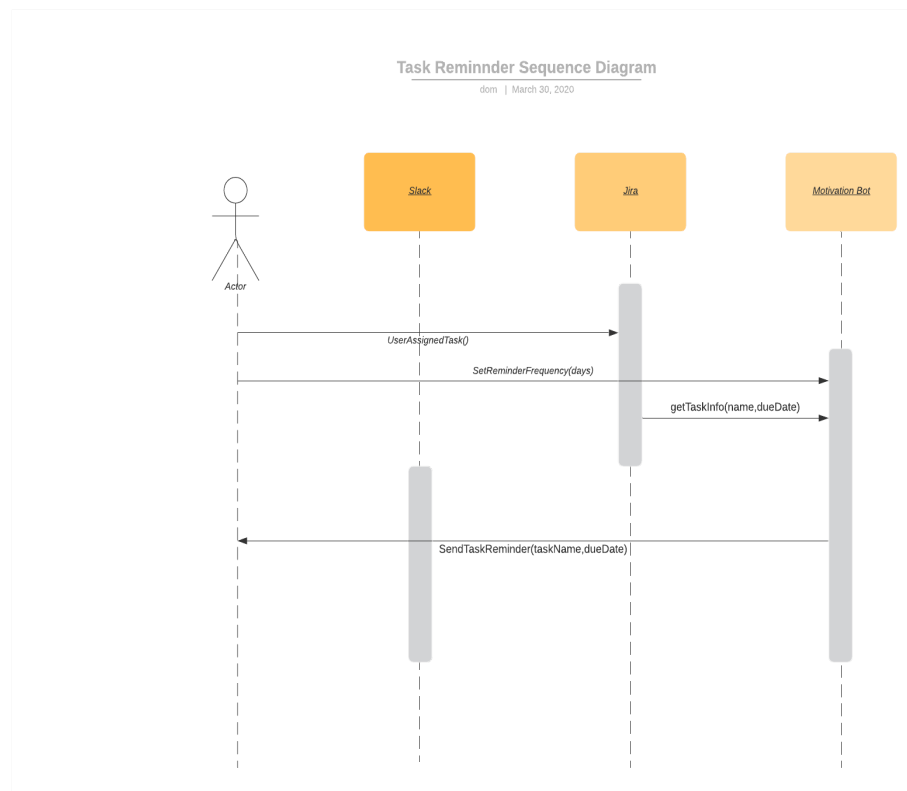
[S3] Bot will send a reminder message at the set frequency or a "Assignment late. Reset expected time".

4 Alternative Flows

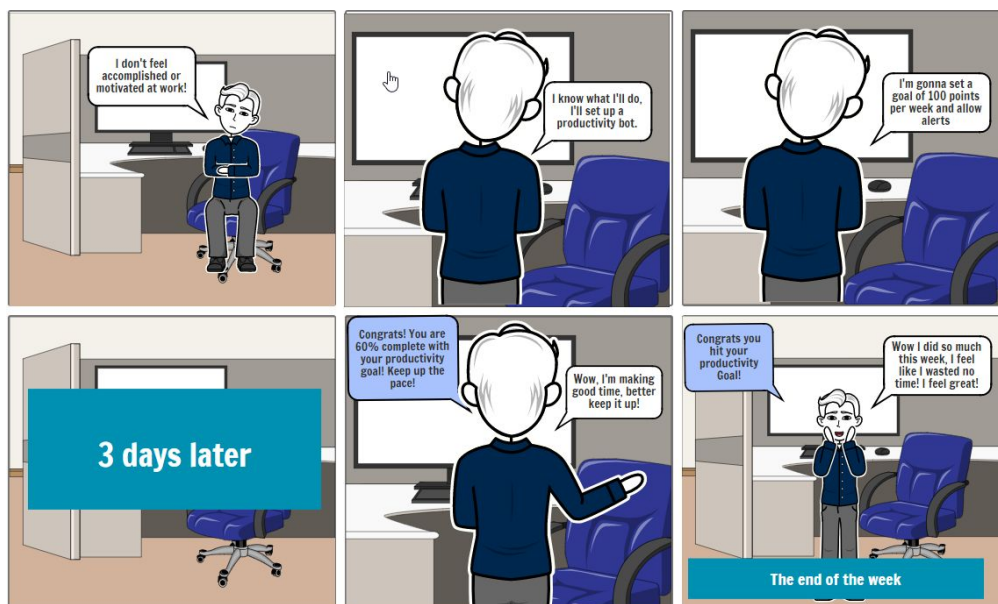
[E1] task has no due date

Design Sketches:

- sequence flow mockup



- Storyboard



Architecture Design:

Diagram 1 - Process Flow

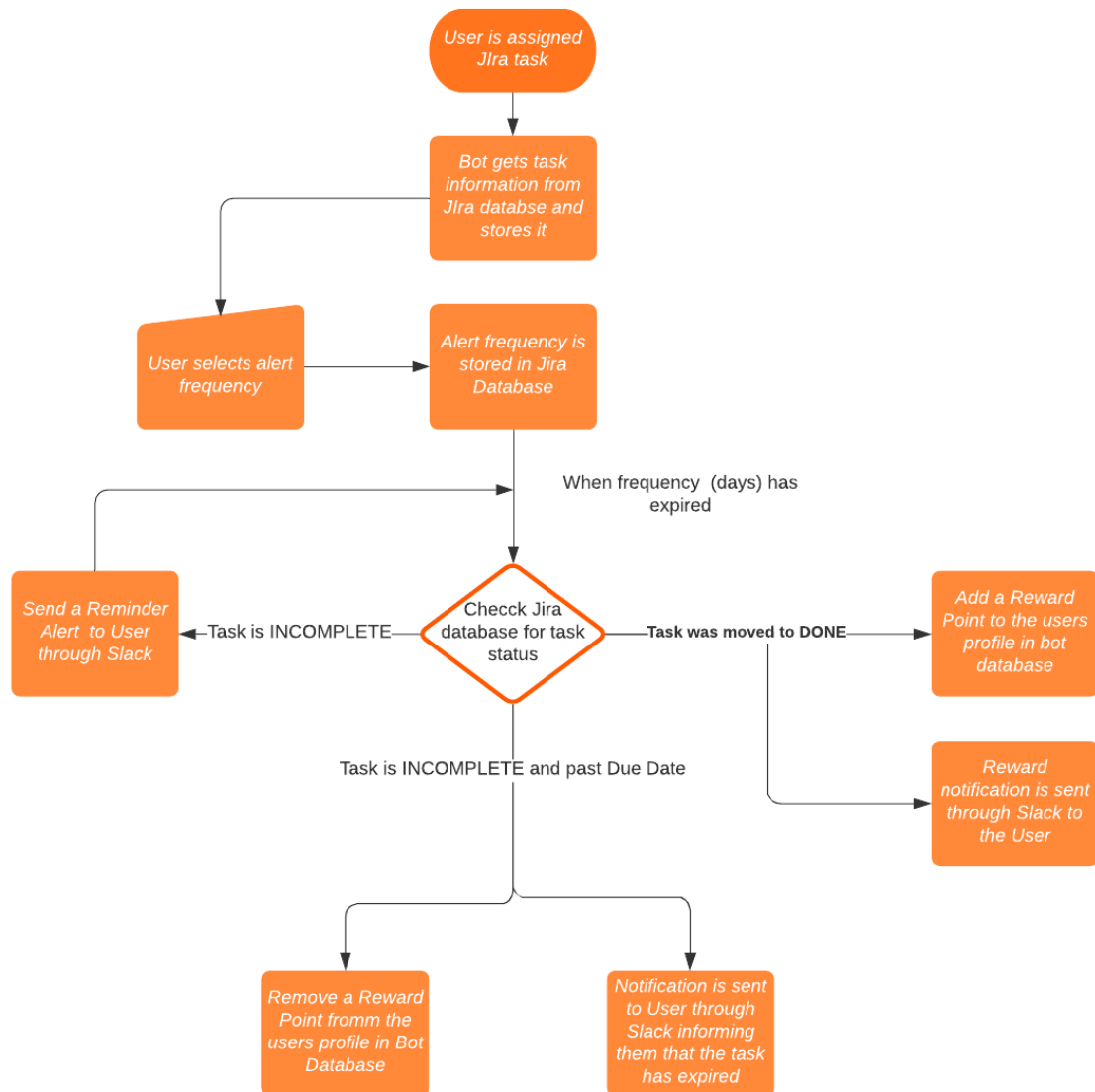
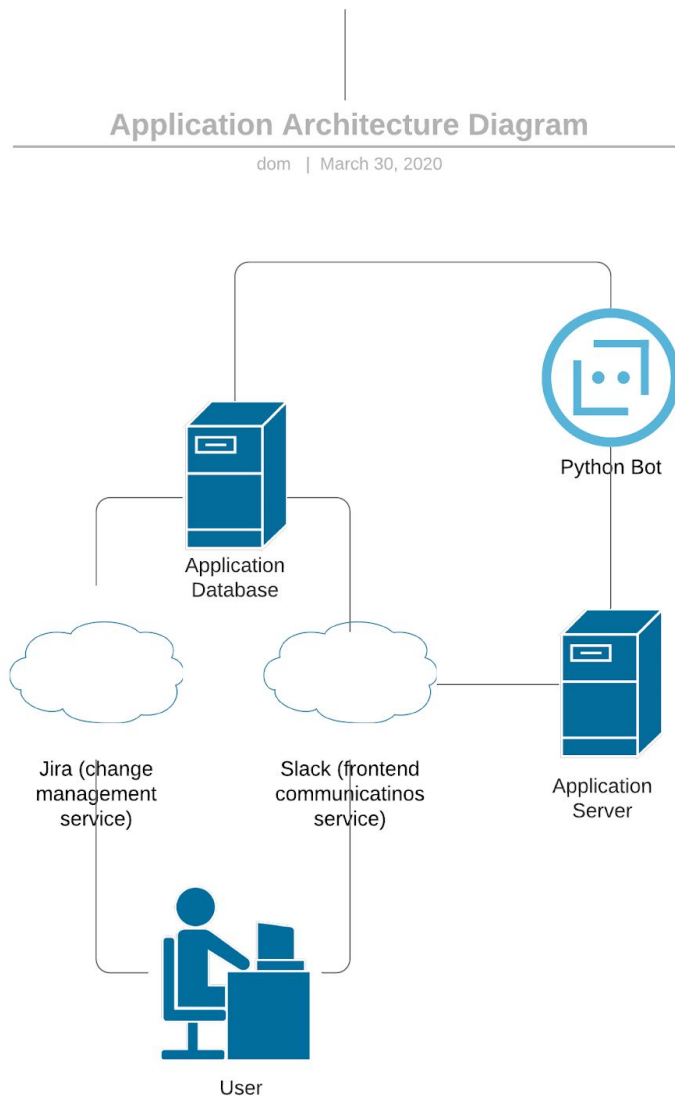


Diagram 2 - Systems Architecture



Application:

Platform - Python, DB

Application - Main Bot Service

Application Server - Web server to host bot application and handle requests/data

Data Storage:

Application Database - stores information gathered from Jira/Slack as well as from interactions via the application

Third Party Services/Interfaces:

Jira (or other change management service) - service that we're pulling user information and tasks/etc from

Slack (or other communications service) - service that we're pulling user information and sending reminders/etc to