



IBM TODO App Project

Requirements Document

Software Engineering Project (CSU33013 & CSU22013)

14th February 2022

Brian Whelan (3rd year)

Adriana Hrabowych (3rd year)

Tom Roberts (3rd year)

Nadia Abouelleil (3rd year)

Arshad Rehman Mohammed (2nd year)

James Merrins Pryce (2nd year)

Liam Reilly (2nd year)

1. Introduction

Overview - Purpose of system

The aim of this project is to create a containerized TODO application from scratch. The application will strive towards helping people organize their day to day tasks in the form of a TODO application. The application will list a user's tasks and allow the user to add or delete tasks in their list so that they never forget what they are meant to be working on.

The client, IBM, has expressed that the focus of the project should be less about the application itself, but the process of employing formal software development and design principles frequently utilised in the computer science industry. For example, using proper version control, working on both backend and frontend at once, testing every update, etc.

Scope

- A containerized, fully tested, TODO application which allows the user to add and delete tasks
- The app should provide a clear and easy to use UI
- Static software analysis and automated testing
- Automatic deployment to Red Hat OpenShift

Objectives and success criteria

The majority of the success criteria will be based on how effective our team uses proper software development standards, both learning and then being able to build from this experience. The application, though, will be built through the following stages:

1. Setup testing and other tools required to build applications
2. Create a rudimentary containerized application that connects to a database
3. Further develop both the application and database to support additional functionality

Ultimately, our CI/CD pipeline must successfully build and deploy our application to Red Hat OpenShift.

Definitions, abbreviations

Standardization: This is a document or file format that is used by one or more software developers while working on the same program. Software standards enable the interoperability of programs written by different developers.

Self-Contained/Containerized: When a program/application is contained within a dedicated separate environment in an operating system that contains only what is needed by the program/application to run.

CI/CD: Continuous Integration/Continuous Deployment

CI/CD Pipeline: Automates the steps in the software development process. The pipeline builds code, executes tests (CI), and deploys a new version of the application in a safe manner (CD). Automated pipelines eliminate any manual errors, provide developers with standardized feedback loops, and allow for rapid product integration.

References

Red Hat OpenShift Developer Sandbox - <https://developers.redhat.com/developer-sandbox>

Red Hat CodeReady Containers - <https://developers.redhat.com/products/codeready-containers/overview>

Local development with OpenShift - <https://developers.redhat.com/blog/2019/09/05/red-hat-openshift-4-on-your-laptop-introducing-red-hat-codeready-containers>

Deploying to OpenShift using GitHub Actions - <https://cloud.redhat.com/blog/deploying-to-openshift-using-github-actions>

2. Current system

There is no current system for this project and it will be developed from scratch. There are a variety of other projects which implement similar CI/CD pipelines from which we will base our project on.

3. Proposed System

Overview

Our proposed system will be a containerized TODO application. The primary focus of the project will be in building a software development pipeline as well as employing collaborative software development practices.

We will be setting up a GitOps CI/CD pipeline in which pushes to the project's GitHub repository will trigger automated testing, packaging and deployment of the application.

The application itself will enable a user to add TODOs to the application which will be stored in a persistent SQL database. Users will also be able to delete TODOs, mark them as complete and sort/filter them. The design and features of the application are at the discretion of the development team.

Functional Requirements

Minimum Viable Product

- The user should be able to add TODOs to their TODO list

-
- The user should be able to remove TODOs from their TODO list
 - The user should be able to mark TODOs as complete

Additional Goals (time-permitting)

- The user should be able to sort/filter the TODOs in some meaningful way
- The user should be able to login to the app in order to save their TODOs
- The user should be able to add a due date to their TODOs
- The user should be notified when their TODO is approaching its due date

Non-functional requirements

Application

- The application should be downloadable as a release from GitHub
- The TODO data should be stored in a persistent SQL database

GitOps CI/CD Pipeline

- Any pushes to the GitHub code repository should result in triggering automated testing, packaging and deployment of the application
- The pipeline should include code scanning, unit test coverage, and static analysis stages
- The CI/CD pipeline should automate the deployment of the application to OpenShift
- Red Hat Quay should be used as the container registry

Agile Software Development

- The project will be built incrementally using an agile approach, with weekly sprints and scrum meetings
- Project backlog and weekly sprints should be recorded on GitHub kanban board
- Code should be reviewed by at least one other team member prior to merging with the main branch
- At all points, the main branch of the repository should contain a working system which pass the automated tests
- Progress will be demonstrated to the client on a bi-weekly basis and feedback will be used to determine upcoming sprints

Security

- The build containers need to be based on the Red Hat Universal Base Image with up-to-date security patches

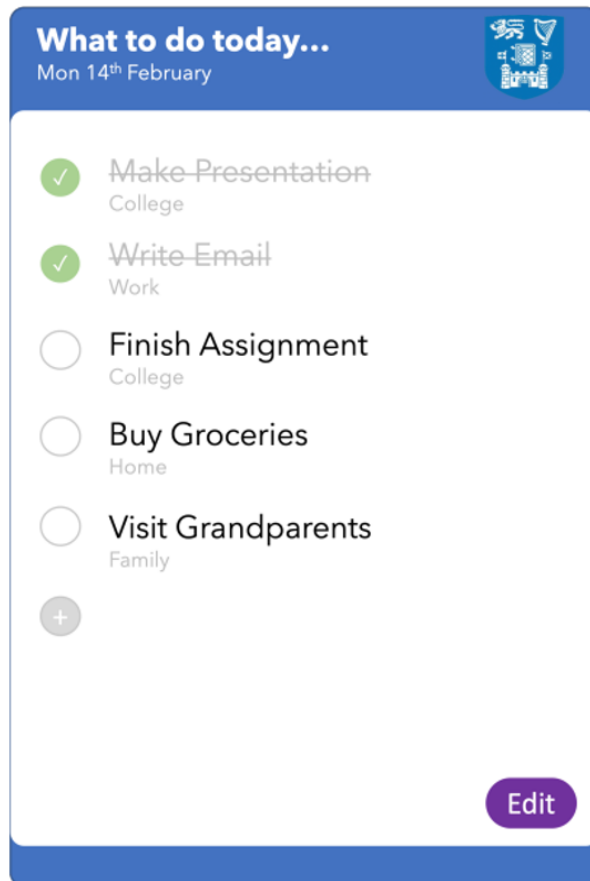
Documentation

- Instructions should be documented for deploying application to OpenShift
- There should be a GitHub pages of the project (similar to this: <https://rht-labs.com/enabement-docs/#/>)
- There should be a markdown file (README.md) documenting the project
- Code should be commented where necessary

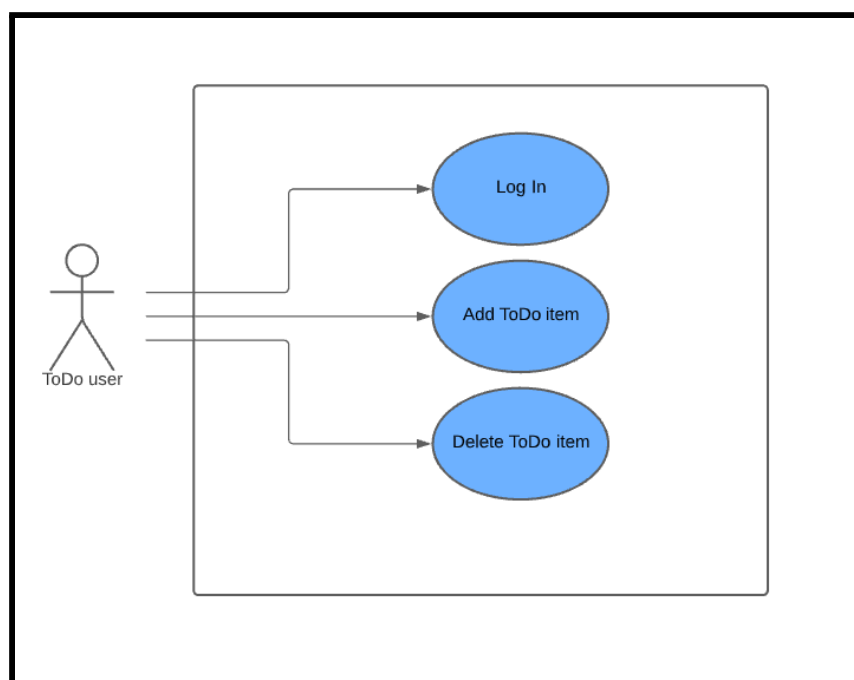
- There should be a final article/LinkedIn post on completion of the project

4. System prototype (models)

User interface mockups



Use cases (including text narratives)



Name: Log in

Participating Actor: ToDo user

Entry Condition: The user has the app installed.

Exit Condition: The user successfully logs in.

Normal Scenario:

1. The user opens the app.
2. The user enters their log in details
3. The user logs in to their account

Alternative Scenario:

1. The log in details were incorrect
2. The account does not exist

Name: Add ToDo item

Participating Actor: ToDo user

Entry Condition: The user is logged into their account.

Exit Condition: The user creates the item successfully.

Normal Scenario:

1. The user logs into their account.
2. The user clicks the "Add item" button.
3. The user enters the ToDo details.
4. The user adds the item.

Alternative Scenario:

1. No more items could be added as too many exist.

Name: Delete ToDo item

Participating Actor: ToDo user

Entry Condition: The user is logged into their account and has at least one ToDo.

Exit Condition: The user deletes the item successfully.

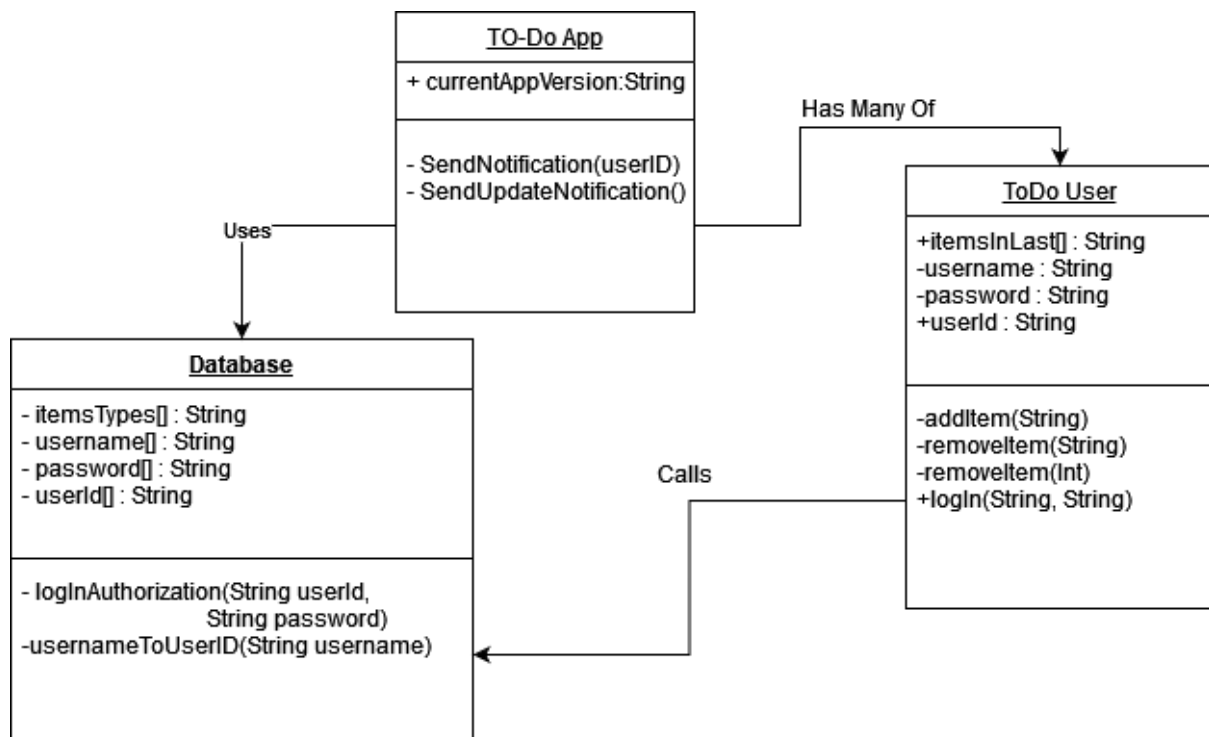
Normal Scenario:

1. The user logs into their account.
2. The user clicks the "Delete" button on the specified ToDo.
3. The user deletes the ToDo.

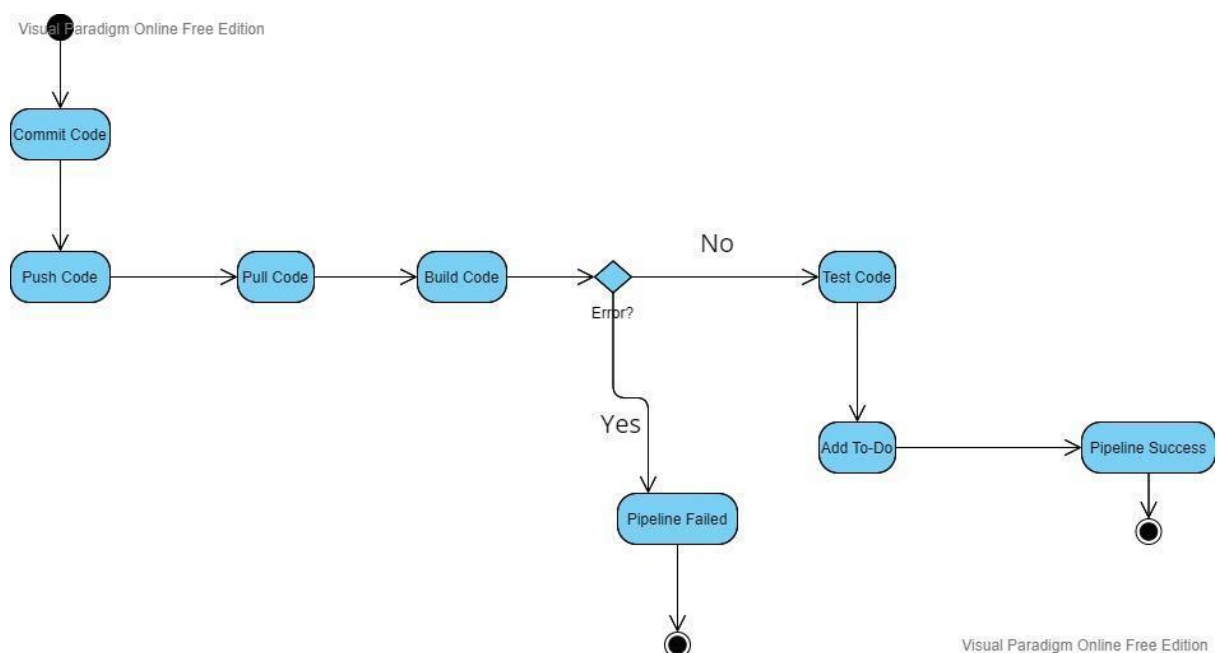
Alternative Scenario:

1. The user has no more ToDo items to delete

Object model



Dynamic model



Client Sign-off

Criveti Mihai

14:11 (1 hour ago)

to me, Panpan ▼

Hi Brian, looks good, **approved** and reflects the changes we discussed on the call!

Do you need anything else this week?

This counts as the required approval of the requirements document.

Thanks!

Regards,

Mihai Criveti

CTO Cloud Native and Red Hat Solutions, Academy of Technology and AoT Red Hat Technical Council Leader, STSM
Certified [IBM x 15](#) | [CKAD CKA CKS](#) | [Red Hat Certified Architect Level 2](#) | [AWS x 4](#) | [Azure](#) | [TOGAF](#)
IBM Consulting - Hybrid Cloud Solutions Center, Hybrid Cloud & Services Solutioning
<http://ibm.biz/openshift-solutioning> <https://w3.ibm.com/w3publisher/global-cloud-solutioning-center-process>
<https://www.ibm.com/w3-techblog/ibm-cloud-solutioning-centers>