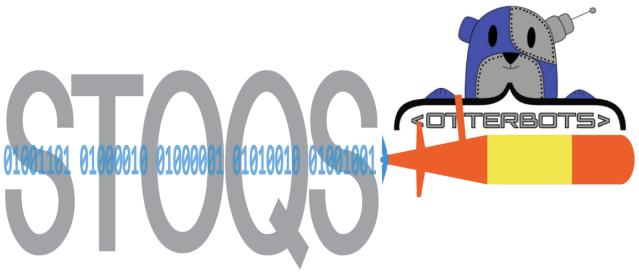

Capstone Project: Spatial Temporal Oceanographic Query System (STOQS)

Client: Mike McCann

Monterey Bay Aquarium Research Institute

Project by Otterbots

Kevin McNulty, Nadia Rahbany, Juli Shinozuka and Andrew Shiraki

CST489: Capstone Project Planning

California State University: Monterey Bay

Faculty Advisors: Cassandra Eccles, Brian Robertson, Bude Su

Summer 2023

Executive Summary

The Project's Purpose

The goal of this capstone project is to build a modernized Spatial Temporal Oceanographic Query System (STOQS) development environment which not only updates the environment but also integrates the use of VS Code to reduce the technical knowledge needed for those using the software. The STOQS open-source software is located on a GitHub repo: https://github.com/stoqs/stoqs.

Goals/Objectives

- Create base project with Cookiecutter Django
- Get Docker containers to run with STOQS models, views, templates, and loaders copied into base project
- Use up-to-date version of PostgreSQL (PostgreSQL 15 or greater)
- Get STOQS unit and functional tests to run without error

The Population Affected by the Project

The STOQS software is used by MBARI and other organizations who wish to learn more about our oceans. Since the software is open source, it has the potential to affect people all over the globe for use in scientific research and private interest.

Anticipated Outcomes

It is our hope to complete this project, but at the very least we hope to make a meaningful contribution to the purpose of our project. We understand that we are working with some technologies that were not directly taught in our curriculum, but feel confident in our ability to adapt. The deliverable for this capstone project is the updated STOQS development environment Docker container that can be published on GitHub.

Table of Contents

Introduction

Project Description

The Otterbots have partnered with Michael McCann at the Monterey Bay Aquarium Research Institute (MBARI) to contribute to the Spatial Temporal Oceanographic Query System (STOQS) project. The researchers at MBARI document and study the health of the World's oceans. They survey the oceans with a variety of manned and autonomous vessels which collect large amounts of data with various sensors. STOQS is an open source software package which allows researchers to access, visualize and analyze this data which contributes to better understanding of our oceans.

Problem

The STOQS software was created in 2010 using mostly vim and accessing a postgres server running on the users machine. Per Mr. McCann's github issue (https://github.com/stoqs/stoqs/issues/1274); the project has been updated to run in a Docker container "but the development environment has not been customized to use the Django development server. This makes it difficult to quickly develop, test, and roll out new changes to the software." There is also the need to update dependencies and technologies, such as the PostgreSQL to a version 15+.

Solution

The Otterbots will be supporting the STOQS project by helping to "pay the technical debt" accrued over the years of the projects existence by creating a base project with Cookiecutter Django and updating legacy components to use the latest software packages.

Environmental Scan/Literature Review

The development platform MATLAB is used for data analysis and numerical computing. It was initially created as a simple terminal application, but over time it grew to incorporate separate windows for graphics, editing, and other features, making it simpler for non-programmers to use. The Monterey Bay Aquarium Research Institute uses the Spatial Temporal Oceanographic Query System (STOQS), a geographic relational database technology, to manage data from observational missions. It manages in situ measurements from stationary platforms and offers efficient access to data from mobile platforms. STOQS, like MATLAB, provides users with a tool to visualize, and manipulate data in many ways necessary for performing complex simulations and mathematical computations.

Stakeholders

The stakeholders involved in the proposed project include end users, developers, and clients. End users, including scientists and researchers, benefit from the advanced capabilities of the STOQS in exploring and analyzing vast and diverse oceanographic data. They gain insights and a comprehensive overview of measurements in space and time, with the ability to focus on specific parameters or instruments. Developers stand to benefit from the project by creating an improved development environment, enabling faster development, testing, and deployment of new features. Clients, such as the non-profit organization supporting the STOQS, benefit from an enhanced software program that provides scientists with valuable tools for data exploration, leading to potential advancements in oceanographic research.

Ethical Considerations

Several ethical considerations are relevant to the capstone project. It is important to ensure that the STOQS remains accessible to a wide range of users, including those with

disabilities or varying levels of technical expertise. Designing the user interface and implementing features with accessibility in mind will help promote inclusivity and equal access to the system's functionalities. Additionally, the project should address privacy and data protection concerns, ensuring that the collection, storage, and analysis of oceanographic data adhere to ethical standards and relevant data protection regulations.

Legal Considerations

The capstone project should address potential legal considerations to ensure compliance. From a legal perspective, it is crucial to respect copyright permissions and licenses when incorporating existing STOQS components into the project. Implementing appropriate data security measures, obtaining user consent, and establishing data retention policies will help ensure legal compliance and protect the rights and privacy of users.

Project Goals and Objectives

The primary goal is to modernize the STOQS development environment by porting existing code base into a new project, a dockerized cookiecutter-django web application.

Goals	Objectives
#1 Create a base project with cookiecutter-django and docker.	● Learn about cookiecutter-django, its configurations, install, and create a base project. ● Learn about docker and create an environment project. ● Create a base development environment project with cookiecutter-django and docker.
#2 Become familiar with the technology stack and file structure of the base project.	● Successfully run a small django project, with PostgreSQL database if possible. working outside the base project then inside the base project. ● Successfully run the same django project inside the base project created previously. ● Modify the development environment if needed during this exploration.
#3 Run the stoqs project in the base cookiecutter-django project previously created.	● Get familiar with the stoqs file structure and make a plan to port. ● Port the STOQS project into the created development environment. ● Troubleshoot until the project successfully runs. (This is the most difficult step.)
#4 Update PostgreSQL and other dependencies if possible.	● Find outdated dependencies and make sure the client wants them updated. ● Update PostgreSQL, if not done already. ● Troubleshoot any issues.
#5 Complete the project by having a working development environment in cookiecutter-django for the stoqs project that passes all required tests.	● Run and pass all tests included in the base STOQS software that was ported into the development environment. ● Troubleshoot any issues.

Final Deliverables

The deliverable the client requested is access to the group GitHub repository for the capstone project. The repo will provide valuable information and documentation about the development process, challenges, and final development environment created during the capstone project This will provide insight for future improvements to the STOQS program.

Approach/Methodology

Throughout the project, an iterative and incremental approach can be followed, allowing for continuous feedback and improvement. Weekly group meetings foster communication, progress updates, and issue resolution among team members. Additionally, weekly meetings with the client to ensure alignment with expectations, feedback incorporation, and a shared project vision. This approach can be adapted from Agile methodologies to facilitate effective collaboration among team members and adapt to any unforeseen challenges or changes.

Research and Familiarization:

- Conduct research on the current state of the STOQS project, including its existing development environment, technologies used, and challenges faced.
- Review the GitHub repository (https://github.com/stoqs/stoqs) to gain an understanding of the project structure, codebase, and dependencies.
- Explore the Cookiecutter Django framework (https://github.com/cookiecutter/cookiecutter-django) and its capabilities for creating a modern development environment.
- Learn about Docker containers and their usage in software development.

By following this approach, the team aims to create an updated STOQS development environment using Cookiecutter Django, Docker containers, an up-to-date PostgreSQL version, and successfully run all unit and functional tests without errors.

Timeline/Resources

Detailed Timeline

Major Stages	Description	Start Date	End Date
Milestone 1	Project Setup	May 31, 2023	Jun 6, 2023
Milestone 2	Testing and Development	Jun 7, 2023	Jun 13, 2023
Milestone 3	Integration of STOQS	Jun 14, 2023	Jul 4, 2023
Milestone 4	Update to PostgreSQL	Jul 5, 2023	Jul 11, 2023
Milestone 5	Testing & Troubleshooting	Jul 12, 2023	Jul 25, 2023
Milestone 6	Documentation & Launch	Jul 26, 2023	Aug 8, 2023

Milestones

Milestone 1: Project Setup

- Create a base project using Cookiecutter Django.
- Set up the development environment with Docker.

Milestone 2: Testing the Development Environment

- Test the development environment by porting a small Django project.
- Identify and make necessary changes to finalize the development environment.

Milestone 3: Integration of STOQS Components

- Copy STOQS models, views, templates, and loaders into the base Cookiecutter Django project environment.
- Ensure Docker containers run successfully with the integrated STOQS components.

Milestone 4: Update to PostgreSQL 15+

- Update the project to use the latest version of PostgreSQL (15 or greater).
- Resolve any compatibility issues or dependencies related to the PostgreSQL update.

Milestone 5: Testing and Troubleshooting

- Run STOQS unit and functional tests to ensure they execute without errors.
- Troubleshoot and resolve any issues or bugs encountered during the testing phase.

Milestone 6: Documentation and Launch

- Document the development process, challenges, and solutions in the project repository.
- Finalize the updated STOQS development environment and prepare it for launch on GitHub.

Resources Needed

Software: PostgreSQL, Python, Django

Dev. Tools: Cookiecutter Django, Docker, Visual Studio Code (VS Code) + Extensions

Collaboration Tools: Slack, Gmail, Zoom, GitHub

Hardware: A computer system with enough processing power, storage space, 4GB RAM , and a dependable internet connection.

Platform

To complete the project of modernizing the STOQS development environment, the chosen platform is a combination of Cookiecutter Django and Docker. The selection of Cookiecutter Django and Docker as the platform for this project was driven by the compatibility with the STOQS project requirements. Cookiecutter Django provides a solid foundation for building Django-based web applications, while Docker ensures a consistent and reproducible

development environment. Together, they offer a modern and efficient platform for updating and modernizing the STOQS development environment.

By leveraging the benefits of these platforms, the team aims to enhance the development process, facilitate rapid testing and deployment of changes, and ensure a seamless experience for developers working on the STOQS project.

Risks and Dependencies

The STOQS project is a full stack application running in a virtual container. Each layer depends on the layer beneath it to properly function. The cascading dependencies of this project add some additional risk. Failure to complete a task has the potential to disrupt the entire project. At a high level, our project consists of 5 major components which are:

1) Database - PostGIS

2) Map Server

3) STOQS - Django framework and utilities + html js

4) Docker - Containers to build development environment

Out of these major components, it appears the areas of highest risk are the STOQS module and the Docker container.

While Docker is a widely used software platform, no group members have any experience with Docker or any container software and depending on how steep the learning curve is, a significant amount of time may be required to get the members up to speed. We consider this a significant risk because the measurable success metric placed on this project states "the project runs in a Docker container."

The STOQS module is the main component of this project, it connects users to the database and map server and is the component that most developers will be modifying while

contributing to this project. STOQS is written in Python and leverages the Django library to serve the webpages, query the database. The software has been in development for many years and appears to have become quite large and sophisticated. Similarly to Docker, getting the team up to speed on such a complex piece of software can be a challenge, and an identified risk for project success.

Testing

The STOQS repository includes unit tests which test the framework with Selenium and test database loading and querying with Django and the built in utils. The testing will include the included unit tests in addition to an end to end test.

Testing Requirements:

1. STOQS unit tests all pass within the Docker container (white box)
2. Users are able interact with the container as expected (black box)

Team Members

Members: Kevin Mcnulty , Nadia Rahbany , Juli Shinozuka , Andrew Shiraki

Division of labor (including clear roles and responsibilities):

In order to address the challenges of learning a new tech stack while simultaneously working to complete a project in a relatively short time period. It may be beneficial for the team to divide the learning responsibilities amongst members. This project requires all members to be proficient users of Docker and Django/Cookiecutter. The initial, albeit naive, plan is to have two members focus on Docker and the remaining two members focus on Django/Cookiecutter. Ideally these paths will converge for milestone #2 where the skills learned in the discovery phase can be quickly shared with the rest of the team. As the development method we have laid out most closely resembles an agile development cycle, planning more than a few weeks out may be

futile. Instead of adhering to a rigid plan. We have planned the milestones and will adjust weekly responsibilities as necessary during the development process.

References

McCann, M.; Schramm, R.; Cline, D.; Michisaki, R.; Harvey, J.; Ryan, J., "Using STOQS (The spatial temporal oceanographic query system) to manage, visualize, and understand AUV, glider, and mooring data," in Autonomous Underwater Vehicles (AUV), 2014 IEEE/OES, pp.1-10, 6-9 Oct. 2014 doi: 10.1109/AUV.2014.7054414

Moler, C. (n.d.). *A brief history of Matlab.* MATLAB & Simulink. Retrieved May 2023 from https://www.mathworks.com/company/newsletters/articles/a-brief-history-of-matlab.htm.