ELABORATION REPORT

Created By

Jose Guillen Santos and Wynne Wu

COMP 3220
Object Oriented Analysis and Design
School of Computer Science

University of Windsor

July 16th 2020

1 Coding Repository

For our coding repository, we used GitHub. GitHub is a popular version control and software development hosting platform. The GitHub project page can be found here and it hosts all of our major code for the project. There are 5 public repositories that should be visible. A documentation repository that will be mentioned later in this report, a repository for the Windsor Data Java API, a repository for the sites domain that hosted through GitHub pages, and two other repositories that hold the major CKAN dependencies for our open data portal. We have separated the repositories in a manner that distinguishes between major elements of the project.

The **ckan-work** and **datapusher-work** repositories host the code for the site's HTML, the site's CSS, the Python backend, and the CKAN API which handles some key functionality for our site and our Windsor Data Java API. The domain repository is temporarily being hosted on GitHub until we can reconcile some port forwarding issues with Apache2 and Solr. Then, the **windsor-data-java-api** repository hosts any code regarding the creation of a Java API for the Windsor Data Portal. As of right now, there is a master and experimental work branch to help implement version control. Within these branches, there are some test applications that were used to test the functionality of the Jackan API. With this tool, we will be able to provide high level functionality for developers interact with the data

2 Version Control

In regards to Version Control, we used Github to manage our coding and documentation repositories with different branches. With this, the team can add to a separate preview branch, and then, once we have a consensus on the design, we will merge to the master branch. This can be seen most clearly with the documentation repository and Windsor Data Java API repository.

A specific example of how we used version control was with updating the designs of the Windsor Data Java API. A member would start drafting the UML class diagram and upload it to the documentation repository under preview, then, once it was confirmed by other team members, it would be merged with the master branch.

3 Documentation Repository

The documentation repository can be found here. Some of the documentation you can find at this repository includes but is not limited to the Phase Plan, Use Case Diagram, and the Iteration Plan for Phase 2. We have also added documentation regarding the build for the Windsor Data Java API.

This repository was especially helpful tracking documentation and giving team members access to them which proved fruitful when we needed to reference the documentation for our build or to keep track of requirements of the use case.

4   Bug Reporting Tool

The Bug Reporting tool that's being used for this project is through Monday.com. The team is using a bug queue template that's being offered in order to manage and track individual bugs, their current status, where the bug originated from, their priority, and submit date. Specifically, the team has used this tool to track bugs with the user login, and issues with the package install of CKAN. For example, on a local build of the site, there was an issue with an admin user being unable to login because they were being redirected to a different site. This issue was tracked, handled, and updated on the Bug Queue on Monday.

This tool was also helpful tracking issues with the package install of CKAN. There was an issue with the Solr schema from the package install from CKAN. Again, Monday helped track, handle, and update the team on the progress of the build. Overall, the tool is helpful in keeping the team updated with the ongoing development of issues with the open data portal. Furthermore, Monday is a very agile and flexible tool. It has the potentiality to integrate with GitHub which would allow our team to seamlessly integrate issues tracked by GitHub users straight to our Monday Bug Queue. This emphasis on agility and flexibility is the main reason why this tool was selected among others.

In regards to viewing the board, we are using free student accounts for Monday which **does not allow the feature for shareable boards**. So, there will be screenshots of the Bug Queue board below and available to view here, and if you wish to look at the board, this is the link to become a team member. We've only enabled viewing access for team members invited through this link, and you will need to use your uwindsor email to sign up for Monday in order to view the board.
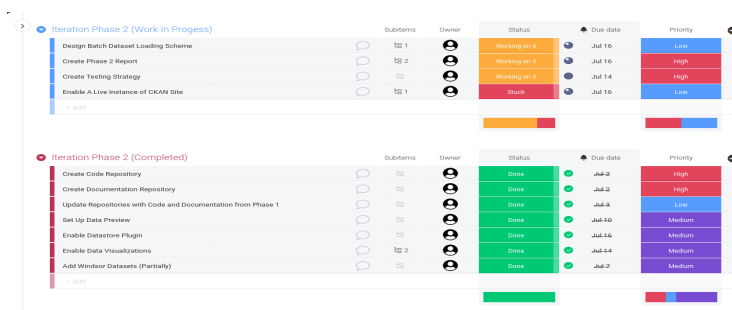

Figure 1: Bug Queue Board

5   Project Management Tool

Task tracking was again done with Monday.com. This tool uses a Kanban system to help us track our project tasks and deadlines. The major reason we chose this for our task tracking was the flexibility of the tool and its ability to keep the project agile in the future. The potential to create autonomous updates to the board was one these agile features on Monday. This could potentially create a more efficient work flow in the future, as more issues and tasks may arise as the project continues to grow.

A specific example of the tool's use and usefulness is its ability to keep the team on track with this reports requirements and to track the feature implementation for the Windsor Open Data Portal. Tasks can be assigned and be given specific due dates, as per the phase plan, to ensure that the project is progressing at a timely pace. This ensures work parity and efficiency.

Again, we are using free student accounts for Monday which **does not allow the feature for shareable boards**. So, there will be screenshots of the Project Management board below and available to view here, and if you wish to look at the board, this is the link to become a team member. We've only enabled viewing access for team members invited through this link, and you will need to use your uwindsor email to sign up for Monday in order to view the board. Note that if you already became a team member to view the Bug Queue Board you don't need to register again, you should be able to view both boards by signing up once.

Figure 2: Project Management Board

6   Testing Strategy

A major part of our testing strategy was to implement Unit Testing. A part of this strategy was to use JUnit to make our own testing process very agile for the future. However, we focused on another part of the unit testing strategy. We chose to be very selective with the packages and libraries that we would incorporate within our project.

For example, we selected Jackan in order to implement our Windsor Data Java API because of its excellent documentation. One of its features was that their package included Unit Testing programs that can be seen in our **windsor-data-java-api** repository here. With these unit testing programs, we can liberally change their API as needed because we have the test cases in place to make sure that the functionality remains unchanged.

We've also added test programs in the Windsor Data Java API repository. Specifically, these test cases are helpful when implementing the Jackan API. The test cases guarantee a synchronous programming experience between team members by ensuring the package works properly on everyone's local machine. These test programs can be seen here.
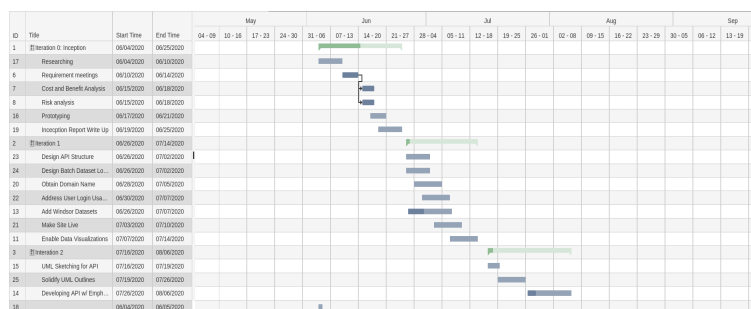
7 Phase Plan

Figure 3: Phase Plan Diagram

The figure above is our phase plan and can be seen in more precise detail in our documentation repository here. The high level objectives to be accomplished during this first elaboration phase includes:

1. Designing an API Structure

2. Designing a Batch Data Loading Scheme

3. Obtaining a Domain Name for the Site

4. Fix Issues with Admin Login

5. Adding Windsor Datasets to the Site

6. Enabling Data Visualization with Site Data Sets

7. Making The Site Live and Public

With these high level objectives, our team gets one step closer to achieving our goal to provide Windsor stakeholders with a state-of-the-art open data tool. Each objective has the purpose of adding value to stakeholders or places us in a better position to add value.

With the first objective, we want to begin the process of creating a quality API tool for our Windsor developers that want to access the city's data. The goal here is to create a workable outline for the API by the end of this phase in order to begin its implementation in the next phase.

With the second objective, we want to ease the process between updating the datasets. Consequently, the dataset would be constantly up-to-date which is essential to the stakeholders. The objective here is to have a rough outline so that we could start to implement the scheme in the next phase.

The third objective brings us closer to being able to present this tool to the stakeholders. With our own domain, we begin the process of making a live instance of the site that we can present to our stakeholders.

The fourth objective, however, allows us to add an administrator user to the site with the ability to edit datasets, add datasets, delete, tinker with the site's aesthetics through changing the CSS, and handle the organizations on the site. This would allow for the city of Windsor to easily maintain the site which provides long term value to its stakeholders. As of the start of the phase, there's been redirection issues when attempting to sign in. Overcoming this major obstacle gets us closer to achieving stakeholder satisfaction.

Once we complete the fourth objective, the fifth objective is to get the team to add datasets to the site and start to experiment with other ways to add value to them. The sixth objective, is a specific way to add value to them. By enabling data previews and starting to implement light data visualization elements, we continue to meet the requirements that the stakeholders have outlined for the team in inception.

The seventh objective is the most challenging and ambitious. As of the start of this elaboration phase, all instances, prototypes, and proof of concepts have been hosted privately on a local machine. Due to all the dependencies required for the functionality of CKAN site build, it is a bit complicated to create a live and public instance this early in the process. If we can further the process, this would be a step in the right direction.

8   Resulting Product

Before we discuss the resulting project, we should mention that we were unable to complete two of our high level objectives—the second and the seventh. The reason

being is that the phase plan was decided while we initially had 3 members in our group. However, one of the members decided to leave the project, so this effectively dropped our team's velocity by a third. Therefore, with only two members, our team's velocity was not fast enough to complete all these objectives. Furthermore, the loss of a team member, in conjunction with our lack of experience with the tools, also limited our team's velocity. For example, our team has limited experience working with datasets, Python, CKAN, networking issues, version control, unit testing, and, etc.

So, when issues arose, it took us a significant amount of time, due to the learning curve, to solve them. This limits the team's velocity which in turn limits our ability to complete our objectives. This specifically affected our ability to make the site live. The team is inexperienced with networking issues. So, when we came across port issues with the site, it would take a significant amount of time for our team to solve this issue. So, the team concluded that our time was better spent finishing other objectives and implementing them on a locally hosted site. These are the main reasons we were unable to make the site live and why were also unable to complete the designs for a batch loading scheme. It is simply because our team's actual velocity was significantly less than our expected velocity for reasons out of our control.While we could have changed our phase plan during the current phase, it is important to our stakeholders that we still attempt to stay on track for our initial plan, even though we have fallen just slightly short.
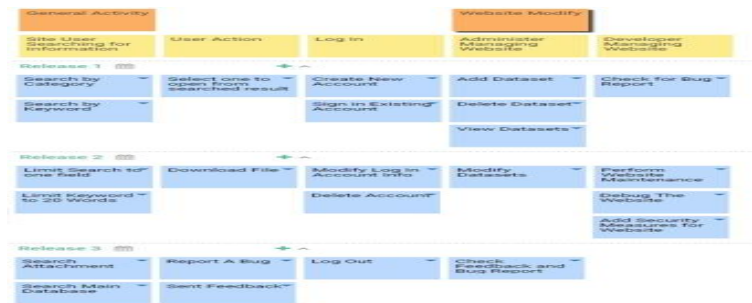

Figure 4: User Stories Diagram

Now that we've discussed our missing objectives, we can discuss the prototypes and diagrams created in order to satisfy the objectives. In regards to objective 1, the team has created a user-stories-diagram and a class diagram for the Windor Data Java API which can be seen above and below. These diagrams generate a solid foundation for the API's scope and a clear idea of how we should proceed to implement the API in the following phase. With the user stories diagram we've outlined clear requirements for the API, and with the Class diagram, we have created an idea of how we are meant to implement these features. These diagrams can be seen in greater detail in

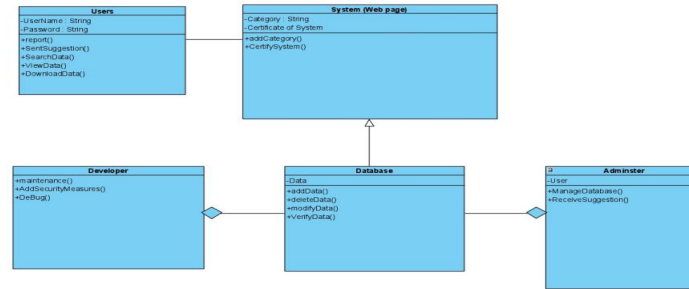the documentation repository located here and here.

Figure 5: API Class Diagram

As we mentioned in the **Coding Repository** section we currently have our domain hosted on GitHub Pages and can be viewed here. There is nothing in this repository because all the resources for the CKAN site are within the CKAN repository ***ckan-work***, and the domain repository is just a temporary location for the domain to be hosted on until we reconcile the porting requirements needed for Solr and Apache2 which are dependencies of the Windsor Open Data Portal site.
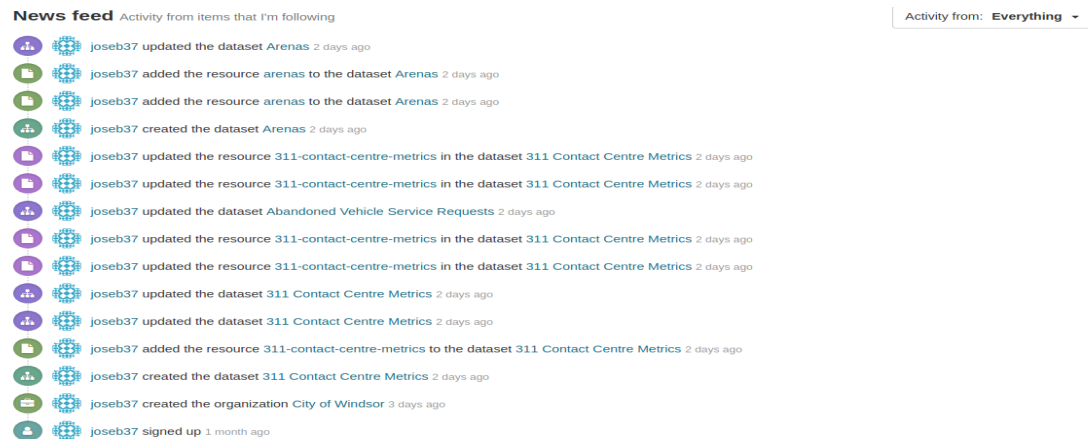
Figure 6: Proof of Concept, Admin User and Features

Now for the last 3 requirements, the team extended upon the site prototype displayed in the inception report. From that prototype we were able to address the login issue, and create an administrator account capable of adding datasets, editing and deleting datasets. Evidence can be seen above, and in the documentation repository here. The screenshots are proofs of concept that we can build a site with a logged in admin

user, who can add new datasets that were added straight from the old data portal site. In the figure below or in this repository you can view the proof concept of adding specifically Windsor datasets.
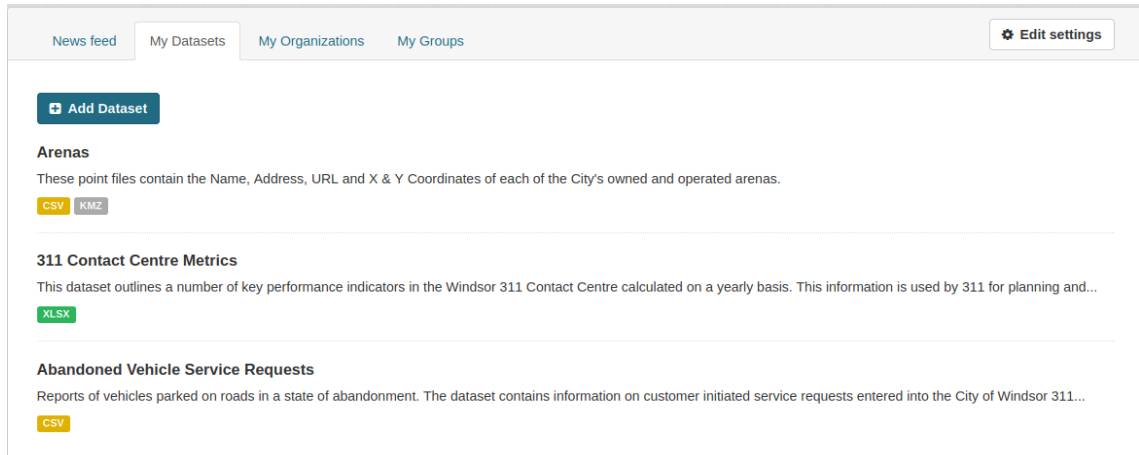

Figure 7: Proof of Concept, Adding Windsor Datasets

We further added the ability for data visualization elements which are demonstrated in the screenshots below and be found in the documentation repository here. This was again by building on the CKAN prototype we had built in the inception phase. We enabled the DataStore plugin for CKAN which allows us to display data visualization. These data visualizations include previews for pdf, jpeg, xls, png and gif file types. You can also enable graph different types of simple graphs such as bar graphs, scatter plots, etc. These visualizations are only available with XLS and CSV file types. These visualizations can also be seen below. This objective was a clear success since these types of visualization will create massive amounts of value for our stakeholders. Furthermore, the fact that we were able to store these visualizations on the CKAN DataStore, means that these datasets can easily be queried by developers since JSON links have automatically generated for each dataset once they are uploaded for the DataStore. This can be seen with the given repository link.

Figure 8: Proof of Concept, Data Visualization w/ Data Preview
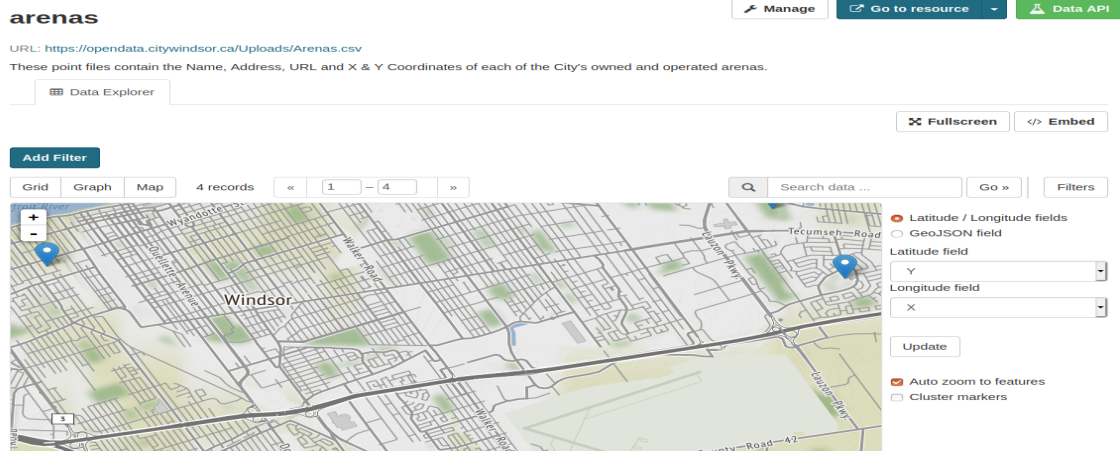
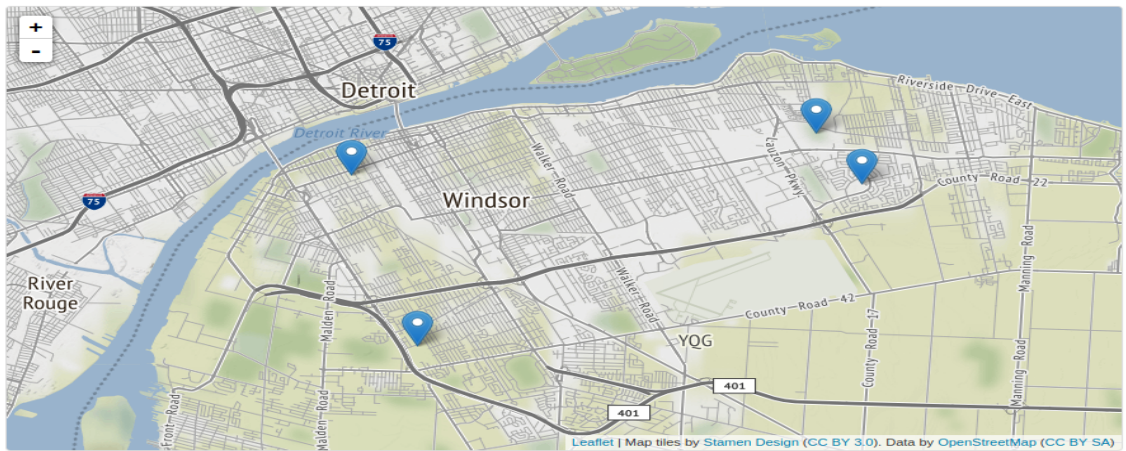Figure 9: Proof of Concept, Data Visualization w/ Maps 1

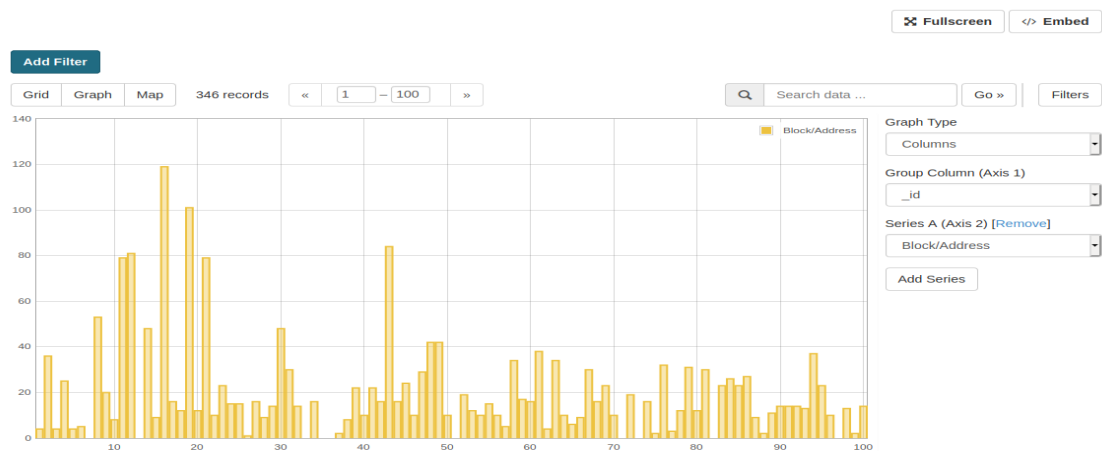Figure 10: Proof of Concept, Data Visualization w/ Maps 2

Figure 11: Proof of Concept, Data Visualization w/ Graphs