CASE STUDY 4: Government of Canada
Group 5

The Government of Canada maintains an open data set that is relevant to Canadians across the country. Includes everything from Agriculture, education, economics, health, labor, and much more. The government has developed an Application Programming Interfaces (API) store to enable people to build new applications and services for Canadians. Using and publishing Open Data and Information | Open Government - Government of Canada

You are to assist the stakeholders in developing the next-generation open data access portal in the form of a live dashboard and user interface, complete with the ability to collect (upload) data, manage it, distribute it in various formats, and deliver high-quality API libraries for developers. Your mission, inspired by what other tools you have seen in these case studies, is to present to the stakeholders related to the particular case study a state-of-the-art tool for open data.

Project Phase 1: INCEPTION - Draw from the case's website(s) any information you need (such as vision, open data plan, etc.) to create a document for inception, addressing all 9 inception artifacts. Be creative and realistic in the features of the open data management tool that you are going to create. DUE: Week 7

Project Phase 2: Elaboration iteration 1. Based on your phase plan in inception, proceed to the first development cycle. Here you need to make sure you have employed all the team collaboration tools including: code repository, version control, bug reporting tool and project management/task tracking for the team (eg. redmine), testing strategy, and documentation repository. Much of this phase is done using UML for all architectural designs from use cases, and any object-oriented programming language (Java preferably) to create code implementations. DUE Week 9

Project Phase 3: Elaboration iteration 2. Here we are going to be focused on design patterns. GRASP patterns are general patterns to help with the development of agile software. GoF are more practical patterns. For this phase, you are to demonstrate at least 5 different patterns in your code clearly explaining/documenting their usage and implementation with full documented UML illustrations of the pattern designs. DUE: Week 10

Keep in mind that it is not a requirement for you to complete the implementation of the entire requirements of this project. We seek the results of two iterations only. This is based on team velocity and justified planning based on the number of tasks to be completed by the team (given its size) in an approximate work schedule of 5 hours per week per person.

Project class Presentation: A 10 minutes demo of your elaboration phase and process outcomes. DUE: Week 11

Final Project Report: Due end of Week 11

We have resources such as gitlab, redmine, and myweb (for documentation) available from help.cs.uwindsor.ca or you can use any other open-source products you prefer.

This is a group project. Teams are to be formed from members of 4 to 5 individuals who are expected to meet regularly and maintain all meeting records.

Use LaTex (e.g. overleaf) for your documents and a proper UML tool (e.g. Visual Paradigm) for your diagrams. No hand-drawn diagrams are accepted. Be sure to use a git type code repository for all your code.

February 14th, 2024: Online meeting via Discord and divided the inception artifacts among the group members. Nadine will reach out to the professor or TA for more details on what we're supposed to do in the project, since it hasn't been made clear.

Artifacts for Inception:

- Vision and business case **(Brett)**
    - High level goals and constraints
- Use-Case model **(Calder)**
    - Functional requirements (10% in details)
- Supplementary specification **(Yize)**
    - Key and mostly non-functional requirements with impact on the architecture
- Glossary **(Nadine)**
    - Domain terminology and data dictionary
- Risk List and Management Plan **(Nadine)**
    - Describe risks (business, technical, resource, schedule)
- Prototypes and proof of concepts **(Hisan)**
    - Clarify vision, validate technical ideas
- Iteration Plan **(Omer)**
    - What to do in the first elaboration phase
- Phase Plan and software development plan **(Arya)**
    - Tools, people, education, and other resources
- Development Case **(Omer/Calder)**
    - Customization of UP steps and artifacts for the project

---

High Level Vision & Business Case

Vision:

We envision an open data access platform operating as a centralized one-stop service for accessing, processing and viewing hundreds of data sets from the Government of Canada. The data portal is to be developed as a means of transparency, equipping citizens, researchers,

businesses and developers with all data resources for many uses. Through setting a federal data usage ecosystem, we believe we can strengthen public services at the national level.

Business Case:

The Government of Canada understands the critical role of open data in building transparent, accountable government processes. Through the establishment of this open data access portal, we aim to address several key objectives:

Enhancing Accessibility: It is our mission to centralize access to a wide range of datasets covering regions in the areas Agriculture, Education, Economics, Health, and Labor. These datasets shall be available to all Canadian residents.

Empowering Decision-Making: Through providing the public, policymakers, and businesses with a complete and timely data, we aim at building up their capacity to take informed decisions and thus implementing necessary changes in different areas

Providing Collaboration: Our goal is to create a collaborative platform for data providers, users and developers to exist so that we can have a vibrant ecosystem that promotes creativity, sharing of ideas, exchange of knowledge and partnerships to solve complex challenges.

Driving Economic Growth: Through opening the doors to value creation from government data, we expect accelerated economic growth, creation of new jobs and triggers innovation in different sectors which eventually leads to the development of the nation.

High-Level Goals and Constraints:

Goals:

- Establish a user-centered and user-focused interface that addresses the needs of the data providers as well as consumers.
- Security, integrity, and privacy of data must be guaranteed through the implementation of strong authentication, encryption, and access control systems.
- Offer user friendly applications for transferring and managing data for providers.
- Support the performance of data sources search, exploration, and extraction with a more advanced level of filtering and browsing.
- Build an effective training module, tutorials, and support resources to guide users to leverage the platform to the maximum.
- Develop a community generated model to keep on enhancing and increasing the range of services and datasets available through the portal.

Constraints:

- Binding to regulatory standards and data governance policies so as to comply with the laws and ethics.
- Limitation in resources such as budget, time, and workload, which may slow down development or do not meet the expectation.
- Compliance with existing government infrastructure standards and systems to promote smooth adoption and connection.
- Scalability and performance issues in case of growing traffic of clients and high rates of data.
- Aim at universality and accessibility by including a broader spectrum of users, that is, individuals with different needs and handicaps.
- Effective collaboration as well as alignment of the stakeholders to mobilize the support and the participation of relevant government agencies, departments and external partners.

Use Case Model

The main use cases for our data access portal are below. We're expressing our use cases as user stories, as we are early in development and don't want to commit to a fully planned format such as operation contracts. In addition, user stories are accessible to stakeholders and will facilitate communication towards finding the optimal set of use cases for the project moving forward.

User Authentication
The user is able to log into their account on the portal, or create one if it does not yet exist.
Preconditions: The user has opened the application and is not authenticated.
The user is prompted to enter their username and password, or click a button to sign up for a new account. If logging in, the program finds the user's account, giving them access to the data portal if they authenticate correctly. If authentication fails, an error message is displayed and they are prompted to log in again. If the user is signing up, they are prompted to enter their email address, username, and a password. A confirmation email is then sent to their email. Upon clicking that link and approving account creation, a new account is registered and they are given access to the portal.

Browse Data Catalog:
The user is able to browse through the available datasets in the data catalog.
Preconditions: The portal is accessible and the user is logged in.
The user navigates to the data catalog section of the main menu. The program displays a list of available datasets with brief descriptions. A search bar is available as well as filtering settings to allow the user to filter and search for specific datasets based on keywords or categories. The user selects a dataset to view detailed information.

View Dataset Details:
The user is able to view detailed information about a particular dataset.
Preconditions: The user is viewing the data catalog and selects a dataset.

The user selects a dataset from the data catalog. The program displays detailed information about the dataset, including descriptions, metadata, file formats, and download options. The user can view sample data or previews of the dataset.

Download Dataset:
The user is able to download a dataset for further analysis or use.
Preconditions: The user has a specific dataset open.
The user selects the download option for the desired dataset. The program initiates the download process in the format selected by the user. Once the download is complete, the user can access the dataset locally.

Upload Dataset:
The user is able to register a new dataset to the portal.
Preconditions: The user is authenticated as a data provider (whitelisted account with the permission to upload datasets).
The data provider navigates to the upload section of the portal. The program prompts the data provider to select the dataset files (locally stored) and provide additional information such as descriptions and other metadata. The data provider submits the dataset for upload. The portal validates and processes the uploaded dataset, making it available in the data catalog.

Search Data:
The user is able to search for specific data within datasets.
Preconditions: The user is authenticated and navigates to the search functionality within a dataset.
The user enters keywords or filters to search for specific data. The program uses their query to retrieve values in the dataset which are displayed locally within the portal. The user can browse through the search results.

Access Data via API:
The user is able to access data programmatically through APIs.
Preconditions: The user is authenticated and navigates to the API section of the portal.
Basic Flow: The user registers new API credentials and accepts a list of terms and conditions for ethical use of the API. The user is able to use these credentials to make API requests to retrieve specific datasets, perform searches and other queries. The portal returns data in the requested format (such as JSON) along with appropriate metadata. The user utilizes the data for whatever application they are developing.

Supplementary specification

- Access Control: The portal must implement robust access control mechanisms to ensure that only authorized users can access the datasets and functionalities. Which means it requires users to set up their user ID, password according to the specified policy or rules.
- Data Management for Administrators: The portal should be able to support efficient data management features for administrators(government employees who are responsible for data management). These features should include the functions such as uploading, storing, retrieving and visioning the data.
- Data Management for Users: The portal should be able to support efficient data management features for users(the public). These features should include the functions such as searching, retrieving and sharing the data.
- Data Visualization: The portal should provide a built-in data visualization tool to allow users to create charts, graphs and maps from the available dataset.
- Monitoring and Analytics: Even though the Open Data Set is available to the public without restrictions, a function that involves monitoring and analyzing should still be included in the portal to prevent crime and harm use of these data.
- User Experience: The portal should prioritize a user-centric design approach, ensuring that users can easily find or discover their desired data and functionalities.
- Guide: There should be a function that includes a detailed description or user manual for how to use this Open Data Access Portal.
- Availability: The portal should maintain a high level of availability, and make sure minimal downtime for users.
- Database Backup: The database backup should be performed daily periodically. Also, the database backup should be kept for a certain amount of time.
- Data Integrity: The data integrity must be maintained throughout all operations.
- Response times: The portal should provide fast response times for all operations.
- Capacity: The portal should be able to store a certain amount of data(10TB for example).
- Scalability: The portal should be able to scale in order to fulfill the growing user base and increasing volumes of data.
- Maintenance: A periodic update routine should be implied by the maintenance team to ensure a bug-free environment and provide further feature enhancements by users' requests.
- Multi-Language Environment: The portal must support the two official languages of the Canadian Government(English and French) ensuring that all interface elements, including menus, labels, and instructions, are available in both languages, and it should also support other languages such as Chinese, Hindi and more based on the ethnic diversity of the Canadian population.
- Accessibility: The program should be able to run on all the mainstream platforms, such as Windows and Mac OS.
- Data Formats: The program should support various data formats commonly used by the public and the government, including CSV, JSON, XML, and others.

- Compliance: Ensure compliance with relevant government regulations and standards regarding data privacy, accessibility, and security.
- Copyright: Verify that all datasets obtained from external institutions or organizations are legally obtained and comply with copyright laws.

Risk List and Management Plan

1. Policy changes: new governments may prioritize other policies over open set data impacting the funding and relevance of data sets.
   Management strategy: search for multiple investors in order to lessen the dependency from one source.

2. User increase : High user demand (this could be caused by many factors) can lead to performance difficulties such as slow downs, timeouts, and no service.
   Management strategy: Implement systems that detect issues right away to keep up with user increase.

3. Competing groups: duplication from groups that are working on a similar project can be a waste of time and money.
   Management strategy: Adopt collaborations with other groups in order to share information to reduce duplication and waste of time.

4. Limited staff: There is not enough members to help with the management of the project
   Management strategy: Prioritize responsibilities and invest in training development courses so that employees retain all necessary information for specific responsibilities.

5. Validated data access: unauthorized access from people who have gained private data which could have an impact on its security, and confidentiality.
   Management strategy: Implement data encryption so that data is translated into different forms. This prevents unauthorized entities from gaining data and only those with authorization are able to read the data.

6. Poor data quality: Having excellent data may not always be the case, poor data such as insufficiencies and discrepancies can cause faulty or misleading results.
   Management strategy: carry out weekly data quality evaluations to detect those insufficiencies and discrepancies etc..

Prototype

Link to a preliminary Figma prototype for the application has been provided. Currently the prototype is a skeleton describing the main UI elements, we are waiting until the elaboration phase to make further design decisions.

link

Plan for first iteration

Week 1-2: Requirements Gathering and Risk Assessment

- Organize workshops with stakeholders to validate and prioritize requirements.
- Develop a comprehensive risk management plan, including mitigation strategies for identified risks.
- **Deliverables:**
    - A finalized list of project requirements.
    - A detailed risk management plan.

Week 3-4: Architectural Foundation and Prototyping

- Design the system architecture, focusing on scalability, security, and modularity.
- Build prototypes for key features, such as data ingestion, user authentication, and the main user interface.
- **Deliverables:**
    - Architectural blueprint and documentation.
    - Working prototypes for critical system components.

Week 5-6: API Design and Development Strategy

- Design the initial set of RESTful APIs for accessing and managing data.
- Establish a development workflow, including continuous integration/continuous deployment (CI/CD) processes, coding standards, and code review practices.
- **Deliverables:**
    - API design document.
    - Development strategy documentation, including CI/CD setup.

Week 7-8: Environment Setup and Initial Development

- Configure cloud services and infrastructure for hosting the portal.
- Begin coding the core functionalities, focusing on the backend and API development.
- Implement basic security measures and data privacy protocols.
- **Deliverables:**
    - Operational development, testing, and production environments.
    - Initial codebase with core functionalities.

Review and Planning

- Week 9: Review and Retrospective
    - Evaluate the progress made during the elaboration phase.
    - Conduct a retrospective meeting to discuss what went well, what didn't, and how processes can be improved.
- Planning for Construction Phase:
    - Based on the outcomes of the elaboration phase, plan the construction phase in detail.
    - Prioritize features and tasks, and allocate resources accordingly.
    - Update the risk management plan based on new insights and the progress made.

Deliverables Summary for the Elaboration Phase

- Finalized project requirements and a detailed risk management plan.
- Architectural blueprint and documentation, along with working prototypes.
- API design document and development strategy documentation.
- Operational environments for development, testing, and production.
- Initial codebase with core functionalities implemented.

Key Considerations

- Ensure continuous communication with stakeholders throughout the phase to validate requirements and prototypes.
- Maintain flexibility in planning to accommodate changes based on feedback from prototyping and early development efforts.
- Focus on identifying and mitigating risks early in the phase to prevent issues during the construction and deployment phases.

Phase Plan & Software Development Plan

To create the data portal for the Government of Canada, we've harnessed a diverse array of tools and technologies meticulously chosen to uphold the pillars of resilience, scalability, and user-friendliness.

In the backend realm, we've tapped into the robust capabilities of Python, opting for the Flask framework, renowned for its versatility and extensive feature set in web application development. To complement our backend infrastructure, we've embraced PostgreSQL as our preferred Database Management System, celebrated for its reliability and sophisticated handling of diverse datasets.

For the construction of RESTful APIs, we've seamlessly integrated either Flask-RESTful or Django REST Framework into our development stack, while simultaneously incorporating OAuth 2.0 to fortify the security of our authentication and authorization processes.

On the frontend front, React.js has emerged as our framework of choice due to its modular, component-based architecture and unparalleled efficiency in managing user interfaces. Collaborative design efforts have been greatly facilitated through the utilization of Figma, a powerful design tool enabling seamless iteration and the seamless incorporation of stakeholder feedback throughout the design iteration process.

Testing has been an integral facet of our development methodology. We've employed Pytest to conduct exhaustive unit testing, ensuring that individual components of our application meet stringent quality standards. For comprehensive integration testing, Selenium has been instrumental in validating the interactions between various system components, while Cypress has enabled meticulous UI testing to ensure an optimal user experience.

Our deployment and DevOps strategies have been anchored in the principles of containerization, with Docker serving as our primary tool for ensuring consistent deployment across a myriad of environments. Additionally, we've embraced CI/CD automation using tools like Jenkins or GitLab CI/CD, streamlining our development workflows and enhancing overall efficiency by automating the build, test, and deployment processes.

By thoughtfully integrating these cutting-edge tools and technologies into our development process, our overarching aim has been to deliver a data portal of unparalleled quality, meticulously tailored to meet the unique needs and specifications of the Government of Canada. Through prioritizing scalability, security, and ease of use, we're confident in our ability to deliver a solution that meets and exceeds the expectations of our stakeholders, while laying the foundation for future enhancements and innovations.

Glossary

1. **Data set**: A collection of data organized in a structured format, often used for analysis or research.
2. **Scalability:** The ability for an application to smoothly be enlarged to accommodate growth, and be able to handle more users/data/throughput without having to be restructured entirely.
3. **API:** Application Programming Interface. An interface allowing a program to easily access resources managed by another program. It defines methods and protocols allowing a system to request data and/or services from another system.
4. **Caching data**: The process of storing frequently used data in an accessible storage area (cache) in order to improve retrieval latency and therefore performance.
5. **Error handling:** The process of detecting, responding to, and managing errors that occur in a system.

6. **Interface**: A connection point where communication occurs between systems or between a user and a system. Each interface has specific rules governing data and request format.
7. **Character encoding:** A system that assigns numerical codes to represent character data. (Examples include ASCII, UNICODE, etc.)
8. **BYTE order mark**: A marker (sequence of bytes) at the beginning of a text file that indicates its encoding scheme and byte order (little/big endian).
9. **JSON:** JavaScript Object Notation. A commonly used data format composed of key/value pairs.
10. **XML:** eXtensible Markup Language. A commonly used data format composed of nested tags/values.