DUY TÂN UNIVERSITY

International School

CAPSTONE PROJECT 2

CMU-SE-451 / CMU-IS-451 / CMU-CS-451

PROJECT PROPOSAL

Version 2.1

Date: 1 - Mar - 2021

EXPERT-DRIVEN SMART DASHBOARD APPLICATION

Submitted by

Vo Van Hoa

Pham Van Tin

Ky Huu Dong

Tran Thi Thanh Kieu

Approved by

Capstone Project 2 - Mentor:

Name	Signature	Date
Binh, Thanh Nguyen __________	________________________	________16 - Mar - 2020

Proposal Review Panel Representative:

Name	Signature	Date

PROJECT INFORMATION			
Project Acronym	EDSDA		
Project Title	Expert-Driven Smart Dashboard Application		
Project Web URL	https://sda-research.ml/		
Start Date	01 - Mar - 2021		
End Date:	02 - Jun - 2021		
Lead Institution	International School, Duy Tan University		
Project Mentor	Ph.D Binh, Thanh Nguyen		
Scrum Master	Hoa, Vo	hoavo.dng@gmail.com	0935.193.182
Team Members	Tin, Pham Van	tinphamvan123@gmail.com	0932.535.175
	Dong, Ky Huu	kyhuudong@gmail.com	0898.246.980
	Kieu, Tran Thi Thanh	thanhkieutran391@gmail.com	0358.583.251

DOCUMENT INFORMATION			
Document Title	Project Proposal		
Author(s)	Team C2SE.06		
Role	[EDSDA] Proposal_v2.1		
Date	01 - Mar - 2021	Filename	[EDSDA] 001 Project Proposal
URL	https://github.com/sdateamdtu2020/SDA-v2.0		
Access	Project and CMU Program		

REVISION HISTORY

Version	Person(s)	Date	Description	Approval
Draft	Hoa, Vo	15 - Jan - 2020	Initiate proposal	x
2.0	All members	07 - Mar - 2020	Finish content of the proposal	x
2.1	All members	16 - Mar - 2020	Update new context diagram & Table compare the different between ver1.0 & v2.0	x

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PURPOSE OF DOCUMENT

- Define the business needs and problems in detail.
- Provide solutions for business needs.
- Provide an overview of resources, schedule, solution, and budget for the project.

The proposal merely introduces the project to the student development teams and provides the up-front information necessary for the team to develop a specification.

1.2. PROJECT GOAL

Expert-Driven Smart Dashboard - means a dashboard that is convenient for users to analyze and review data. It will include several datasets about real-time information of the environment, measurement data of air pollutants... EDSDA will connect and analyze data from multiple sources in ways you've never imagined. Then reveal the insights you've been missing...in just a matter of minutes.

With smart suggestions and an intuitive visual interface, SDA makes it easy for any user to combine data and discover hidden insights in one place...without the usual scripting, coding, and IT hand-holding. With this dashboard, individuals or any subjects can take advantage of environmental data to be able to decide the best relevant policies.

Not only for the environmental expert, but EDSDA also helps the user directly interact with their own data source by importing, automatically cleaning their data, and enabling the user to do everything with their own data in the easiest way.

1.3. BACKGROUND

Our environment is always changing. However, at the current rate of urbanization and industrialization, outside of the natural factors, the change of environment is mainly due to human factors. Emissions, population explosion, industrial solid waste, ... are the main causes leading to negative effects on the global environment. To address this at a holistic level, data analysis and aggregation are the first important tasks to be done.

However, analyzing and aggregating data from many different sources takes a lot of effort and money. To solve this problem, based on our knowledge of big data systems, we have built an intelligent data processing system that can be run on a website-platform with an intuitive and easy-to-use dashboard. This system is a prospective and useful tool for environmental experts and policymakers in Vietnam in particular, and worldwide in general. It will collect, analyze and synthesize data about all the factors that can affect the environment, thereby helping users to come up with quick and accurate solutions to solve problems related to the environment.

In addition, data sources that open the environment in a particular aspect such as area emissions, soil erosion, etc. are often scarce and stored only in the user's local computer. Analytical tools of these types of data often require expertise using quite advanced technology, thus limiting interoperability with data stored in individuals. With this in mind, we built a subsystem where users could bring their data to EDSDA and analyze, aggregate, and visualize their data directly on the dashboard along with a wide range of system visualization tools.

2. PROBLEM DEFINITION

2.1. USER NEEDS

USER STORIES		
ID	**Actor**	**Epic**
1	As an environment expert	I want to drag a data cube from the cubes list and drop it onto the main content board so that I can choose the property of data that I need and visualize them on suitable kinds of charts and maps.
2	As an environment expert	I want to drag some data cubes from the cubes list and drop them onto the main content board so that I can choose the property of data and then connect these data cubes together for data operation and visualize it on suitable kinds of charts and maps.
3	As an environment expert	I want to view the data source of any data cubes so that I can verify the accuracy of the data.
4	As a user	I want to import my own data sources to the EDSDA database so that I can interact with my data directly with EDSA help.
5	As a user	I want to review my data sources before submitting them so that I can check the accuracy of my data.
6	As a user	I want to define the dimension and measure of my data source's data cubes by myself so that I can easily do OLAPing with my data source.

ID	Actor	Epic
7	As a user	I want when dragging and dropping data cubes with the same one or many dimensions, those dimensions will be automatically connected without manual manipulation, thereby saving time manually linking.
8	As a user	I want to publish the charts and data tables that I build on the dashboard so that I can use them for a variety of purposes including researching.

2.2. NON-FUNCTIONAL REQUIREMENTS

Below are the non-functional requirements that are being offered for the system :

- **Portability and compatibility**: The system is operated on a web-based platform and has the ability to run on any web browser.
- **Security**: Users can use the system without the fear of revealing personal information.
- **Availability**: The system can run continuity 24/24 a day.
- **Usability**: The system has a friendly and flexible user-interface and a great user experience.
- **Reliability:** The system has accurate and transparent data, functions that do exactly their job.

2.3. FUNCTIONAL REQUIREMENTS

Below are the functional requirements that are being offered for the system, which are the backbone of the project :

- Import data sources.
- Auto ETL
- Auto RDF Generation.
- Auto linking data.
- Drag data cubes from the cubes list.
- Drop data cubes onto the main content board.
- Connect the data cubes that have the relation between them.
- Merge the cubes to create a new data cube based on connected data cubes.
- Visualize the data cube as a line chart, a pie chart, a column chart, or a geographical map.

3. CURRENT STATUS OF ART

3.1. COMPARE WITH SIMILAR PRODUCTS ON THE MARKET

There are many research topics on data science applied to policy-making. Here are some researches that come from data scientist groups around the world.

1. Data science empowering the public (https://www.waze.com/): The Smart City dashboards in Rio de Janeiro, Brazil, were created to solve problems related to public transportation and traffic. For this, an infrastructure, a dashboard, and a data portal with more than three thousand datasets and seven APIs for real-time data use (www.data.rio) were developed and used by the Center of Operations Rio (COR).
2. Interactive Dashboards (http://www.data.rio/): Using Visual Analytics for knowledge Transfer and Decision Support (Samar Al-Hajj, Ian Pike, Brian Fisher) -A visual analytics dashboard that reflects the needs and preferences of injury stakeholders. The types of visualizations were selected to efficiently illustrate trends and patterns in injury data.
3. Triangulum City Dashboard (https://moovitapp.com/): An Interactive Data Analytic Platform for Visualizing SmartCity Performance (Mina Farmanbar, Chunming Rong).

It can be seen that data science is increasingly weaving and useful in all areas of life, from urban management to public policy management, and help in making decisions. But unfortunately, the world today is facing a vital problem, the environment. However, in today's market, there are very few data analysis platforms to help policymakers make the right decisions about environmental protection.

This is the reason that makes us build a smart dashboard called Smart Dashboard Application - SDA, the first environment data analysis dashboard in the market now. With Smart Dashboard Application, the users can easily choose and integrate any data that are satisfied with their needs and visualize it in many kinds of charts and maps, so that they - can consider the factors affecting the environment in the most comprehensive and intuitive way.

3.2. COMPARE WITH CAPSTONE 1 PRODUCT VERSION

In the first version, Smart Dashboard Application, we provide users with the following basic features:

- A static data warehouse of environmental data cubes.
- A drag & drop method for choosing data cubes to the dashboard.
- A manually linking data cubes method.
- Some visualization tools.

It can be seen that the two biggest disadvantages of the SDA version, are:

- The fixed data warehouse.
- The manual connection method by the user.

First, with the goal of targeting users who are not only environmental experts but also many other related aspects, it can be found that a fixed data warehouse is a rather limited thing, because of the specific aspects data are always updated and stored by experts on their personal computers, so data crawling tools will not be able to access them and enter the SDA system. This would be very detrimental for professionals to use SDA for specific data types. Realizing this, we have built **an intelligent data import system** that enables users to manually upload their flat data onto the system and interact with them as multidimensional data cubes.

The second thing, in the previous version, when users dropped data cubes on the dashboard and linked them together to aggregate data, they were forced to link the data cubes together by drawing connection lines. This takes a lot of time, even for first-time users of the system and for experienced users. To solve this problem, we have provided the system a method to **automatically detect data cubes with the same dimensions and connect them automatically**. This will improve the ability to attract users, the most important thing is to save time and manipulation of users, ensure efficiency and intelligence for the system.

SDA & EDSDA COMPARISON		
	Smart Dashboard Application	**Expert-Driven Smart Dashboard Application**
Data source	Fixed	Dynamic, user can import their own data sources
Import data	No	Yes
ETL process	Manual	Automatic
RDF Data Cubes Generator	Manual	Automatic
Data Cubes suggestion	No	Yes
Hierarchy structure	No	Yes
Dimensions UI	No	Yes
Dimensions & Measures suggesting & selecting	No	Yes
Data Cubes automate linked	No	Yes
Export report	No	Yes

4. ENGINEERING APPROACH

4.1. CONTEXT DIAGRAM

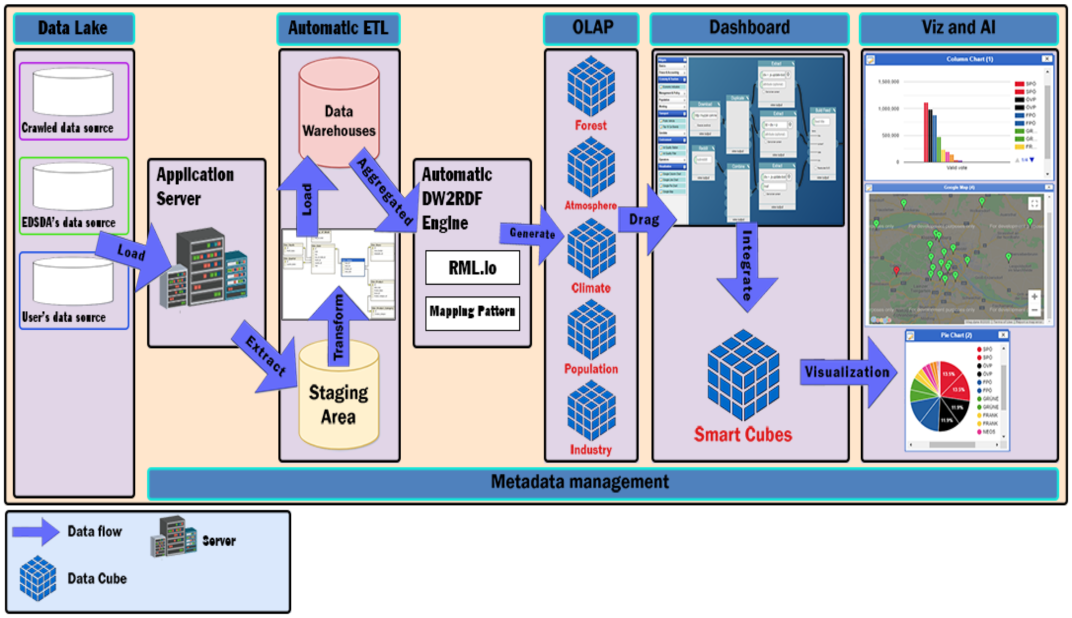

Figure 1: Context Diagram

4.2. PROCESS DETAILS

1. **Data Lake:**
   - Collect from environment open data platforms of the governments and NGO organizations.
   - User imported data sources.
   - Use web crawling techniques to crawl data from related environment websites.
   - Data Format : CSV, JSON, XML.
2. **Application Server:**
   - Check for the changes of the data lake when the user imports new data sources.
   - Handle Data Lake Exceptions if any errors occur.
   - Trigger the Data Lake to Staging Area loading process.

3. Automatic ETL (Extract, Transform, Load)

3.1. Extract:

In this step, we extract structured data after restructuring the raw data from the data lake. Mainly in this step, we focus on exacting data correctly from the structured data from the data lake.

3.2. Transform:

Based on the business's constraints we have to transform the data to be compatible with our process to handle the data for displaying on the dashboard. Getting exacted data, transforming the data, and ready to load into the DW.

3.3. Load:

Loading data into the target data warehouse database is the last step of the ETL process. We load all the data we have transformed into the staging areas of DW and based on the data we have loaded into staging areas we used to load the data from the staging areas to dimension tables, and fact tables.

3.4. Perform automatic ETL:

Performing automatic ETL using tools for automating exacting data, transforming data, and loading data to DW. We use the available platform for scheduling to perform ETL at a specific time in a day. In addition, for performing ETL automatically we also build a data pipeline to make the data flow as we wanted.

4. Automatic DW2RDF Engine:

- Using RML.io with our custom features, it can generate RDF Data cubes automatically when users import new data sources to the data warehouse.
- Suitable with all data sources with a general mapping method.

5. OLAP

In this step, we store the data of the data warehouse as OLAP cubes. And then, for better query performance, data binding, and scalability, in addition to information transparency, we will automate converting OLAP cubes into RDF Data Cubes.

6. Dashboard

In this step, the user can drag any data cubes that appear as items on the sidebar and drop them onto the main content board, then connect between them and use the operator such as statistics merge, geo merge to build a new data cube that matches the user requirement.

7. **Viz & AI**

   This step will perform the data cube which was created by the user with the form they want. It can be a map, a column chart, a line chart, or a pie chart.

4.3. TECHNICAL TO DEVELOP SYSTEM

Main Programming Language: Javascript, Python.

Data Crawling:

- Programming Language: Python, Javascript.
- Database: PostgreSQL.
- Library: Beautiful Soup 4, Selenium.

Data Warehouses:

- Programming Language: Python.
- Database: PostgreSQL.
- Library: Psycopg2, CSV.

Perform ETL automating:

- Tool for ETL process: Talend.
- Schedule for ETL process flatform: Apache Airflow.

Data Cubes

- Programming Language: RDF-Graph, SPARQL.
- Tool for converting from Data Warehouse to RDF Data Cubes Storage: RML.io (with customized phase), MIRROR.
- Network Accessing: RDF-REST API.

Server:

- Programming Language: Javascript.
- Framework / Libraries: ExpressJS (NodeJS), enapso-graphdb-client.
- Operating System: Windows, Linux, macOS.
- Deployment Environment: Google Cloud with App Engine and SQL Services.
- Network Accessing: HTTP methods (POST, GET) via RESTful API.

Client:

- Programming language: HTML, CSS, Javascripts.
- Framework/Libraries: React, Redux, Material-UI, react-dnd, beautiful-react-diagrams, React Charts,...
- Deployment Environment: Google Firebase Hosting
- Web Browser: Chrome, Firefox, Microsoft Edge, Coccoc
- Network Accessing: World Wide Web (WWW), HTTP methods (POST, GET) via RESTful API with Axios.

4.4. THE REASON BEHIND THE TECHNOLOGY STACK

Data Warehouses:

We choose PostgreSQL for the database. It tries to conform with the SQL standard where such conformance does not contradict traditional features or could lead to poor architectural decisions. Many of the features required by the SQL standard are supported, though sometimes with slightly differing syntax or function. Further moves towards conformance can be expected over time. As of the version 12 release in October 2019, PostgreSQL conforms to at least 160 of the 179 mandatory features for SQL:2016 Core conformance. As of this writing, no relational database meets full conformance with this standard.

Data cubes:

We choose to use the RDF model because it allows describing resources (either digital or taken from the real world), by specifying the values of their properties. Thus, an RDF information unit is a triple s p o, with s, p, and o standing for the subject, property, and object, respectively, in RDF terminology. The RDF language is increasingly being used in order to export, share, and collaboratively author data in many settings. For instance, it serves as a metadata language to describe cultural artifacts in large digital libraries and to encode protein sequence data, as in the Uniprot data set. RDF is a natural target for representing heterogeneous facts contributed by millions of Wikipedia users, gathered within the DBpedia data source, as well as for the Linked Open Data effort, aiming at connecting and sharing collectively produced data and knowledge.

Server:

We chose ExpressJS to build the webserver. It's a prebuilt NodeJS framework that makes creating server-side applications simple, fast, and flexible. NodeJS is powered by Google's V8 engine which means it's powerful and can handle a large number of requests without lapsing in dependability. Also, this means that this is a highly scalable choice when you consider the Event Loop which manages all asynchronous operations allowing the program to continue to run as expected without stops.

Client:

We choose React to build the web client. React is developed by Facebook and right now, React and its ecosystems are the most popular stack for front-end developers.

React introduces a new way to interact with the browser: Virtual DOM. We can now build web UI a lot faster than using the old one, the traditional jQuery library. The UI has also become easier to develop and maintain. Besides that, React makes a huge impact on performance, every change on the client-side is now handled by the browser so that it will reduce server load. Together with Redux, React became an amazing tool: good performance, fast development and deployment speed, and a safe future. React has also got a huge and extremely active community that we can get help easily. And React-DnD is a set of React utilities to help you build complex drag and drop interfaces while keeping your components decoupled.

5. TASK AND DELIVERABLES

5.1. TASKS

WBS NUMBER	TASK TITLE
1	Preparation
2	Data Modeling
3	Physical Warehouse Design
4	Initiate Dynamical Data Warehouse
5	Finding Data Sources
6	Data processing
7	ETL Process Validating
8	Automate ETL Setting
9	Expanding database
10	Crawl data
11	Building Data Cubes Importing method
12	Building additional features
13	Common RDF Data Cubes Structure Designing

WBS NUMBER	TASK TITLE
14	Setting up Automate DW to RDF Process Tool & Environment
15	Implementing Automate DW2RDF Process
16	Setting up RDF Data Cubes Storing & SPARQL Endpoint
17	Validating RDF Data Cubes
18	Building SPARLQL_Rest API
19	Building UI
20	Testing
21	Integrate
22	Deploy
23	Release

5.2. DELIVERABLES

No	Activities	Deliverables
1	Project Proposal	Project Proposal Document 2.1
2	Product Requirement	
3	Project Plan	
4	Product Backlog	
5	Architecture Document	
6	Database Design	
7	Interface Design	
8	Test Plan	
9	Test Case	
10	Acceptance Criteria	
11	Sprint Backlog & Burndown Chart	
12	Team Reflection	
13	Technologies Stack	

6. PROJECT MANAGEMENT

6.1. ABOUT SCRUM

Scrum is an agile method, so it follows the principles of Agile Manifesto (see also Agile Manifesto). In addition, Scrum operates on three core values, also known as Scrip Scripps, including Scrutiny, Inspection, and Adaptation.

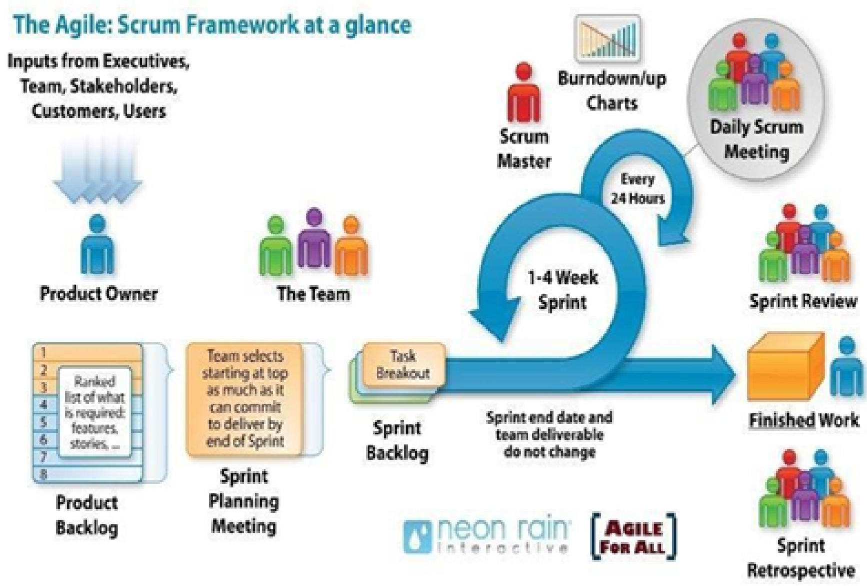

Figure 2: Scrum process

Based on the empirical process control theory, Scrum uses iterative and incremental algorithms to optimize efficiency and control risk. Scrum is simple, easy to learn, and has wide applicability. To be able to use Scrum, we need to understand and apply the elements that makeup Scrum include the core values (also known as the "three legs", or the three pillars of Scrum), roles, Events, and Scrum-specific artifacts.

6.2. WHY SCRUM

- Our team has 4 people
- The project will be continuously horizontally scaled up.
- There is only a short amount of time to finish the project.

So based on those constraints, we decided to choose SCRUM as the project lifecycle.

6.3. TOTAL COST ESTIMATE

COST DESCRIPTION	DETAIL
Salary	Duration: 16 weeks (112 days) Man-hour: 4hours/1day Salary: $2/1hour Persons: 4 members Overtime cost per hour: 1.5$/hour The salary of 1 person: $1000 **Total: $4000**
Laptop	Laptop for each member: $800 **Total: $3200**
Monitor	LG Monitor for each member: $100 **Total: $400**
VPS Engine server	VPS Engine server per month on Google Cloud: $100 VPS Engine server for 3 months: 60$ **Total: $300**
Maintenance	Cost per month: $400
Bugs fixing	Cost per bug: $20
Total	Salary + Laptop + Monitor + VPS Engineer server **Total: $7900**

6.4. TENTATIVE SCHEDULE

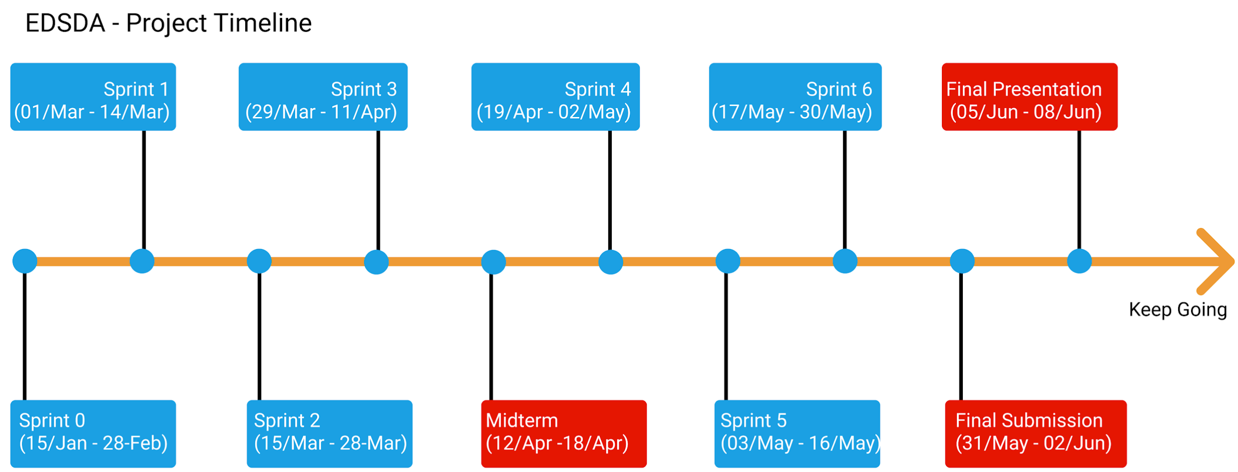

Figure 3: Project Timeline

WBS	Task	Start	End	Days
	Sprint 1	**01-Mar-21**	**14-Mar-21**	**14**
1	Crawl data			
2	Clean data			
3	Initiate RML process			
4	Initiate MIRROR			
5	Initiate GraphDB Database			
6	UI Design			
7	Initiate & config React project			
8	Testing			
	Sprint 2	**15-Mar-21**	**28-Mar-21**	**14**
1	Create DW			
2	ETL Process using tool			
3	Define common mapping for RML & MIRROR			
4	Implement API for importing data, getting data from GraphDB			

WBS	Task	Start	End	Days
5	Building UI			
6	Testing			
	Sprint 3	**29-Mar-21**	**11-Apr-21**	**14**
1	Getting, and Cleaning imported data			
2	ETL the imported data			
3	Describe common schema for RDF Data Cubes			
4	Run RML & MIRROR with sample data			
5	Integrate sample UI & API			
6	Building UI			
7	Testing			
	Midterm	**12-Apr-21**	**18-Apr-21**	**7**
1	Deploy version 0.1			
2	Review documentation			
	Sprint 4	**19-Apr-21**	**02-May-21**	**14**
1	Automating clean data			
2	Schedule for ETL data pipeline			
3	Customize RML & MIRROR			
4	Implement pipeline for automating RDF Data Cube generation			
5	Building UI			
6	Testing			
	Sprint 5	**03-May-21**	**16-May-21**	**14**
1	Handling new importing data to DW(Dimensions, Measures)			
2	Automating ETL for new importing data			
3	Integrate automate ETL & RDF generation			
4	Automating detect dimension & measure			

WBS	Task	Start	End	Days
5	Building UI			
6	Testing			
	Sprint 6	**17-May-21**	**30-May-21**	**14**
1	Automating crawling data from NGO, or any new defined data source.			
2	Cleaning data process for the new defined data source.			
3	Set up VPS for deploying			
4	Migrate GraphDB to VPS			
5	Building UI			
6	Integration			
7	Testing			
	Final	**31-May-21**	**08-Jun-21**	**9**
1	Review documentation			
2	Release / Deploy final version			
3	Presentation			

7. PROJECT CONSTRAINTS

CONSTRAINTS	CONSTRAINTS DESCRIPTION	GUIDELINES FOR ACCEPTANCE
Economic	In terms of cost, since it is a dashboard with not too many screens, the main issue lies in the cost of researching, implementing a fully automated backend system and frontend with smart methods. There is also the cost of renting a server and deploying a server that should be considered carefully	Design cost: Must be around $100. Production cost: Should be under $4000. Maintenance cost: Should be around $500. Operation cost: Should be under $1000
Ethical	Because this software analyzes data with most of the data being crawled, so pay attention to copyright and data security.	All crawl data sources must be open data and be allowed for public use.
Social and Global	The software is targeted to not only environmental experts but also data analysts in many other fields, so widespread popularity is very important.	The product needs to be developed in an optimal and user-friendly way to reach a wide range of users
Sustainability	Need to maintain the continuous operation of the system, so as not to affect the analysis of people	Development and maintenance work must be ensured to take place continuously, when issues are reported, it is necessary to focus on maintenance immediately. Server operation also needs to be ensured not to be interrupted.

8. CONCLUSION

The project will be finished in 13 weeks and divided into 6 sprints (2weeks/sprint), 1 week for midterm review, and 4 weeks for pre-sprint. It promises to be a convenient and useful tool for not only environmental experts and policymakers but also those who love to protect the environment. The confusion about a large data system and reviewing it in a general way to come up with precise planning will no longer be a problem, even for those who don't have technical expertise. The project will contribute to helping our society accurately and effectively fight against environmental damage and global warming.

9. REFERENCE

- Draw tool: https://www.figma.com/
- Software Development Standards for the Guidance and Control Software Project https://sw-eng.larc.nasa.gov/
- General Software Coding Standards and Guidelines https://www.nws.noaa.gov/oh/hrl/developers_docs/General_Software_Standards.pdf
- Scrum and best practices https://docs.microsoft.com/en-us/azure/devops/boards/sprints/best-practices-scrum?view=azure-devops
- The Scrum Guide https://www.scrum.org/resources/scrum-guide
- The ISO/IEC & IEEE/EIA Standard 12207, IEEE standards: IEEE-829 [3], IEEE-1008 [5], IEEE-1012
- Technologies Stack Document: https://docs.google.com/document/d/1DnWvniATIfAxsXl8dq2-REEw9L7tW5cYyz_W0GR6n9U/edit?usp=sharing
- React documentation: https://reactjs.org/docs/getting-started.html
- Redux documentation: https://redux.js.org/
- Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities: http://www.athanasiadis.info/assets/pdf/ems2018.pdf
- Smart cities survey: Technologies, application domains and challenges for the cities of the future: https://www.researchgate.net/publication/333684319_Smart_cities_survey_Technologies_application_domains_and_challenges_for_the_cities_of_the_future
- Interactive Dashboards: Using Visual Analytics for knowledge Transfer and Decision Support: https://www.researchgate.net/publication/299633100_Interactive_Dashboards_Using_Visual_Analytics_for_knowledge_Transfer_and_Decision_Support
- Dashboard technology-based solution to decision making: https://www.researchgate.net/publication/277140671_DASHBOARD_TECHNOLOGY_BASED_SOLUTION_TO_DECISION_MAKING
- Triangulum City Dashboard: An Interactive Data Analytic Platform for Visualizing Smart City Performance: https://www.mdpi.com/2227-9717/8/2/250
- Data Science of the Natural Environment: A Research Roadmap: https://www.frontiersin.org/articles/10.3389/fenvs.2019.00121/full
- IEEE: https://www.ieee.org/
- QBOAirbase: European Air Quality Database as an RDF Cube: http://luisgalarraga.de/docs/qboairbase.pdf

- Enhanced Living Environments: https://link.springer.com/content/pdf/10.1007%2F978-3-030-10752-9.pdf
- An Approach To Publish a Data Warehouse Content as Linked Data: https://pdfs.semanticscholar.org/7913/6c3ad6f91aba1c9d186ede80fb1a1ec54704.pdf