Documentation Software

System Design Document

Prepared For:
Dr. Ghanim AL-Sulaiti

By:
Abdulla AL-Obaidli
Arham Faraz
Waad Sharfi
Yousef Khanfar

September 11th, 2022
67-373 Information Systems Consulting Project

1. Executive Summary

This design document is presented for Qatar Computing Research Institute (QCRI), a non profit multidisciplinary computing research institute founded by Qatar Foundation in 2010.

In this project, we design a content management system using wiki.js which allows researchers within QCRI to view projects, people and products in QCRI using frequently used web browsers. The admin will be able to create,add and delete users,projects,people and products.

In the document, we start with stating the Must/Could functional and non-functional requirements. We then discuss the architectural design and display a UML diagram. Since we are implementing our system using wiki.js, Client-Server architecture is this project's architecture.

According to the specifications, Wiki.js will be polished and updated as part of our implementation strategy. Since Wiki.js makes use of Vue.js, HTML, SCSS, and Javascript, Our team will be employing these technologies to develop the product. Additionally, the team and QCRI will host locally utilizing Windows and MacOS.

The project is fairly simple data-wise, and so there are no major decisions to be made in terms of data design. Projects include both publications and products, and therefore each of these is an entity. Users will mainly be researchers, so a User entity contains an admin attribute to determine if the user can make changes. Furthermore, there is also a researcher entity that must also be a pre-existing user.

Table of Contents

2. Introduction

QCRI consists of numerous departments, none of which can communicate with one another about the projects they are working on. Arabic Technologies, Social Computing, Data Analytics, and Cyber Security are just a few of the departments that make up QCRI. All of these departments perform their research in these specific fields. However, there are times when researchers from these different departments may want to know what is happening in these various projects or may wish to work on a project between the departments. When this occurs, the researchers find it very challenging to collaborate because there is not a website that allows them to view what is being worked on in other departments. Instead, because there are so many distinct topics being worked on at QCRI, the researchers hear about research going on in various departments through word of mouth, which is incredibly inefficient. By addressing this issue, the researchers at QCRI will be able to use their time more effectively, work on projects they may not have known about in the past, and have fruitful exchanges about the research being done in QCRI that may lead to more advancement in other fields.

Our team's objective is to develop a content management system accessible through web browsers for QCRI researchers and project managers to locate research projects they want to collaborate on with other departments, submit their research so the QCRI researchers from all departments can see what is being done, and raise more awareness for the work that is being done at QCRI.

3. Requirements

3.1 Functional Requirements:

View Functionalities:

View	Requirements	User Group	Priority
1 – Project View	1.1 A user must be able to view a project's information (abstract, start, end dates, researchers involved, publications)	Admin/Researchers/Project Managers	Must
	1.2 A user must be able to view all the projects conducted at QCRI		Must
2 – Researcher's View	2.1 A user must be able to view other researchers' information (titles, photographs, projects)		Must
	2.2 A user must be able to view all current products at QCRI		Must

3 -Product/ Outcome View	3.1 A user must be able to view detailed information about a product (description, researchers managing it, related publications)		Must

Table 1: View requirements

User Functionalities:

Functionality	Requirements	User Group	Priority
4- CRUD Operations	4.1 A user must be able to add a new project	Admin/Project Manager	Must
	4.2 A user must be able to update a project status (current, completed)	Admin	Must
5 – Filtering	5.1 A user must be able to filter projects (by date, researcher, or department	Admin/Researchers/Project Managers	Must
	5.2 A user could be able to filter products (by name, researcher)	Admin/Researchers/Project Managers	Could

Table 2: User requirements

3.2 Non-functional Requirements

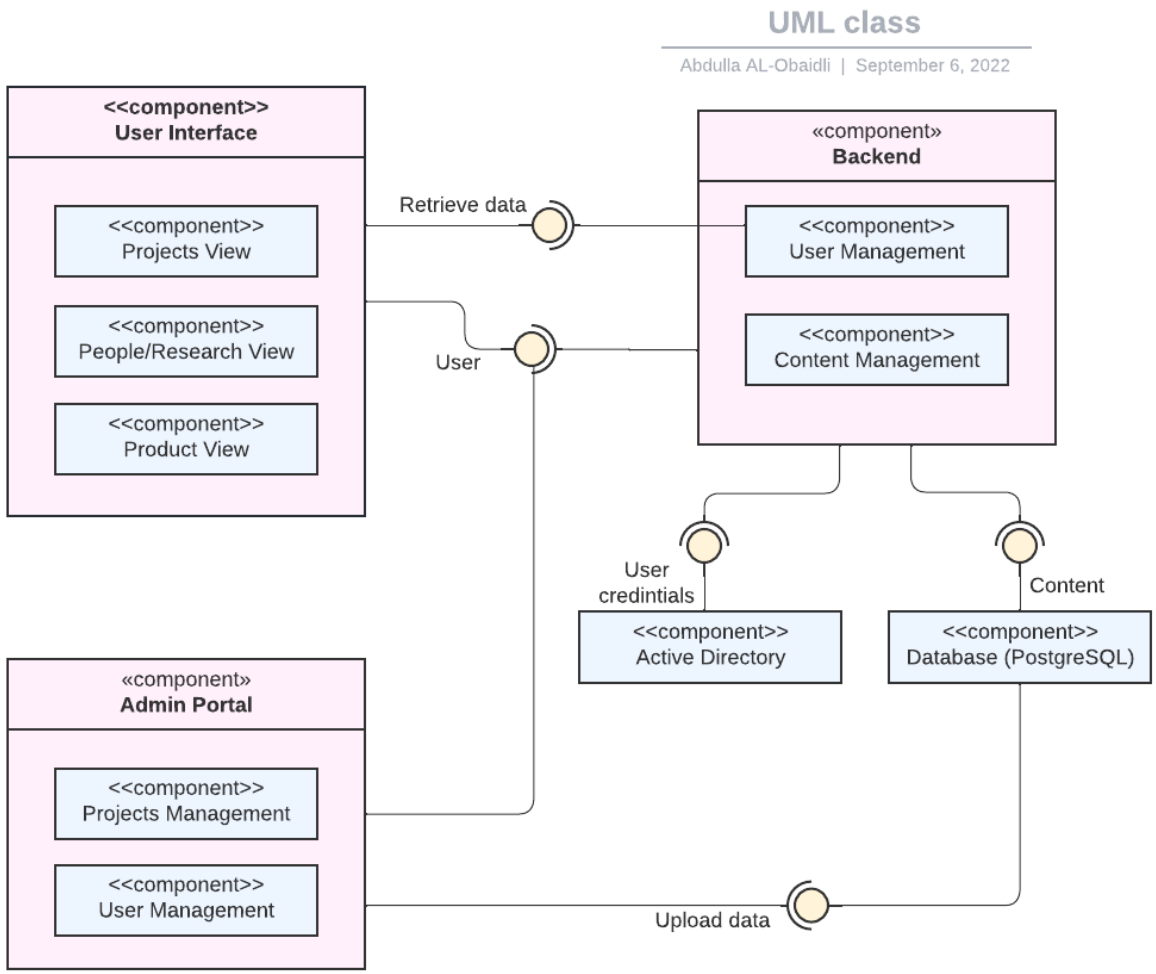
Category	Non-functional requirements	Priority
1- Operational	1.1 The system must be run on web browsers like Safari and Google Chrome. 1.2 The system must run on Windows and iOS.	Must Must
2 - Performance	2.1 The system must respond in 3 seconds or less. 2.2 The system must allow 100 users to operate simultaneously (worst case).	Must Must
3 - Security	3.1 The system must be upon login. (Admin and Project Managers only). 3.2 Authorization is based on roles. Admin and Project Managers only have privileges.	Must Must

4 - Cultural and Political	4.1The system must run in the English language so it can be accessible to all researchers at QCRI.	Must

Table 3: Non-functional requirements

4. Architectural Design

4.1 Solution components and UML component diagram



The wiki.js web application consists of three components, projects view, people/research view and product view. Where the researcher is able to find all the projects as a list or by filter

mechanisms such as by department or author. Moreover, the researcher can read more about the authors rather than the projects themselves and then see the research done by that author. The researcher will also be able to view the outcomes of these researches (e.g publications). Overall, all of these components together guarantee that the software offers any content needs a researcher is looking for.

The admin portal has two components, project management and user management. All of the projects uploaded to wiki.js will be uploaded by the admins, the same goes to deleting those researches or updating them. The same thing goes to users, the admins are the ones who create the users and give the credentials to the researchers, allowing them to access the page.

The backend will also have two components, user management and content management, user management will retrieve information from the active directory to confirm user credentials when logging in. Content management acts the same way when a client would like to retrieve content such as a project overview or information about an author.

4.2 Interfaces and systems integration

The UML presents how all of the components communicate with one another. The backend user management component interacts with the active directory when confirming user credentials by extracting the username and password from the active directory for both the admin portal and wiki.js web application. The backend's content management component extracts the relevant information from the database. The researcher using wiki.js web application will obtain and be able to view such information when desired.

4.3 Architecture decisions

We will be using Client-Server Architecture for this project using wiki.js which would be accessible using the most frequently used browsers such as Google Chrome, Opera, Firefox, Brave and Microsoft Edge.

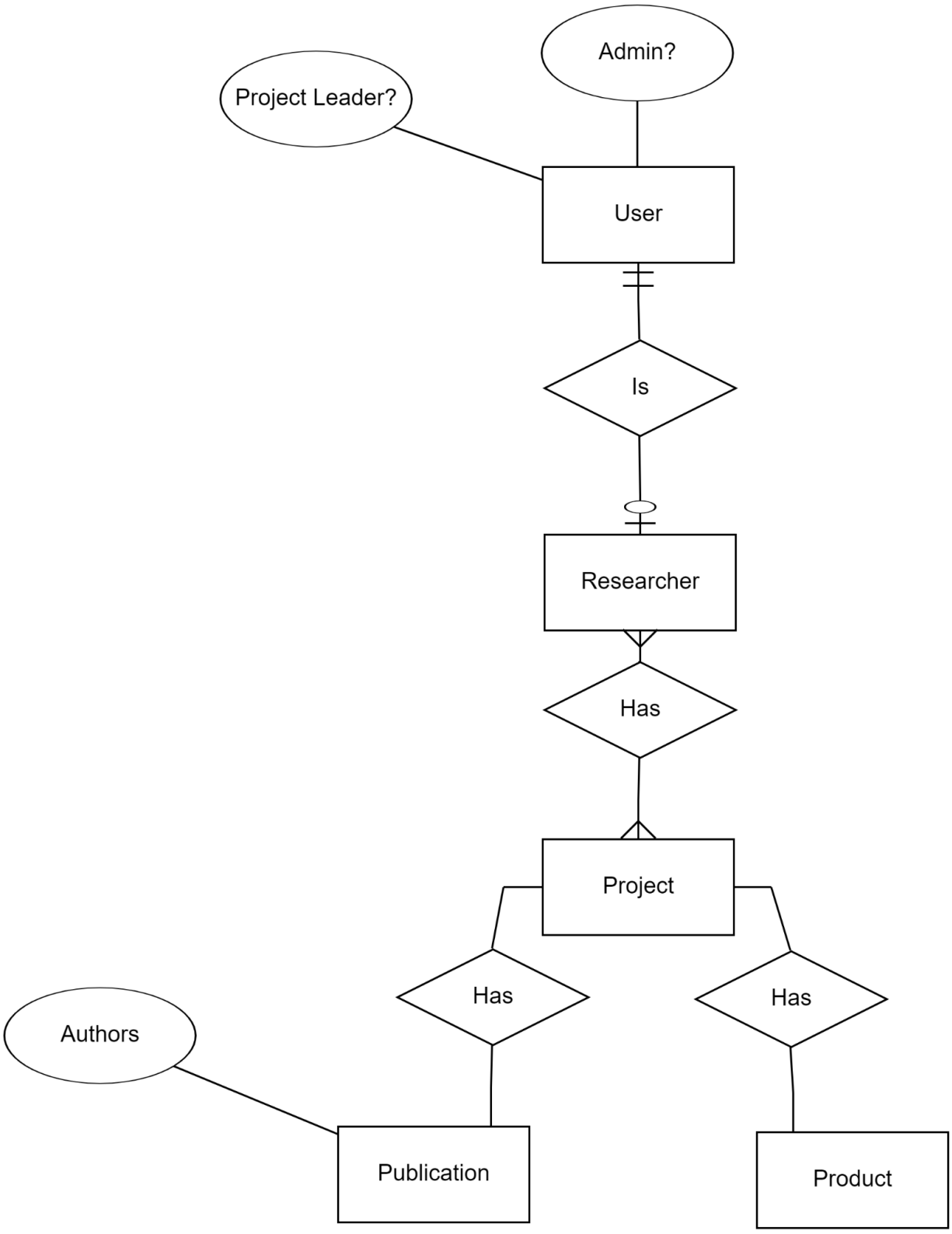
We chose the client-server architecture because it is the most appropriate to our project. Since our clients are researchers who will use the software to read more about projects/authors and publications; whenever they click on a clickable text such as the name of a project, the server responds by showing the relevant data in which the user clicked upon. This architecture style is very easy to adapt to and will be customized to the client's needs to be as efficient as possible to QCRI researchers.

5. Data Design

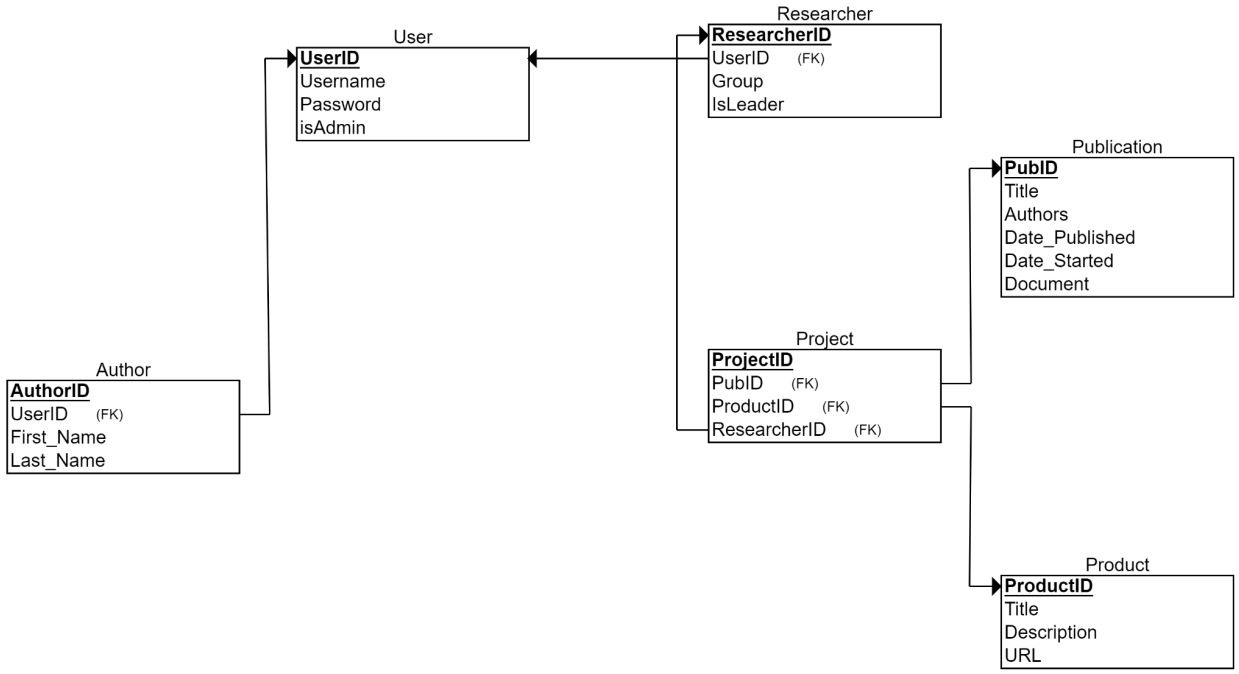
The first entity to bring to focus is the user. The user is an entity that describes anyone who is using the system in place. Whether the user is an admin, project leader, or researcher will be determined via a property within the user entity. The goal of the solution is to facilitate knowledge sharing and collaboration between different research groups in QCRI. The platform must allow for the sharing of publications, products, and authors. Therefore, each of those is also an entity. A product must be linked to a publication, but a publication may not have a product. Furthermore, every author must be linked to an existing user (consult client about this, it's possible that some publications may be partly authored by non-QCRI researchers).

A user does not necessarily need to be an author, but an author must be an existing user. A publication may have multiple products and a product may be a result of multiple publications (consult with client on this). An author can have multiple publications and a publication can be authored by many authors.

5.1 Conceptual data model



5.2 Logical data model



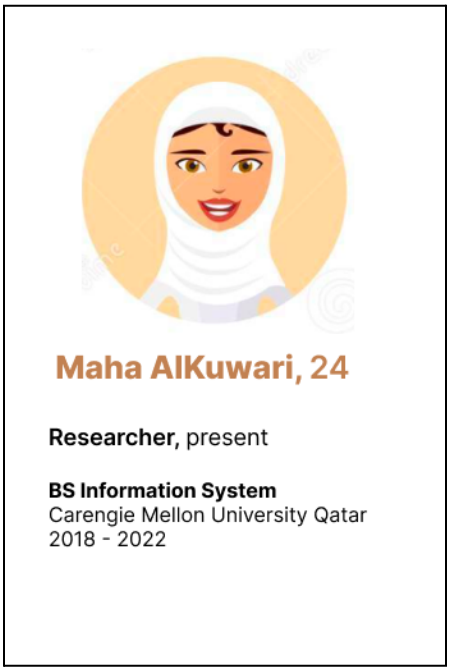
5.3 Data storage and databases

The database will be managed by PostgreSQL, and since the database consists mainly of documents, it will not require large storage spaces. The same computer that will be used to run the server locally within QCRI can be used to store the data. As for data itself, it will be provided by QCRI. Wiki.js currently has a PostgreSQL database that it uses to store all the data for the different pages on the platform. We will be creating a new database, separate from the existing one, that will be used to store the information about the projects, publications, users, etc.

6. User Interface Design

6.1 Understanding the Users

In order to further understand users in QCRI, we have interviewed our client to explain more about our current prototypes and what are the possible design options that would be better for the different users in QCRI.



Maha wants to learn details about QCRI's products.	
"I'm focused on creating new products within QCRI"	
Challenges:	Currently, there is a page for QCRI products, but it only has a brief description about the product. The product description does not include researchers involved or related

	publications. Therefore, it is difficult for a researcher to learn about the owners of the products.
Knowledge:	Very limited knowledge about existing products of other research departments within QCRI.
Tasks:	Arrives at the home page, selects “Our products” in the top bar, selects a specific product, checks detailed information about all products in QCRI. Want to be able to view related outcomes and researchers involved. Want to have easy access to other researchers’ profiles.
Interests:	Interested to be up to date with current products. Want to be able to easily have all the necessary documentation for projects.
Characteristics:	Curious, exploratory, eager to learn new things.

Table 1: QCRI’s persona 1

Mubarak wants to publish a new project.	
“I’m focused on making my team complete tasks on time and produce good quality work”	
Challenges:	Currently, there is no existing system in QCRI for research groups to know about other running projects. The only way to learn about projects is by word of mouth or directly asking each group.
Knowledge:	Understands project management. Has experience in evaluating projects.
Tasks:	Arrives at home page, logs in as a project manager, selects “add project” in the top bar, checks if the project was published successfully. Want to easily publish projects and their relevant outcomes to other researchers.
Interests:	Wants to expand knowledge to his peers. Wants to share new information that could help other researchers in the future.
Characteristics:	Open-minded, deliberate, expects a simple documentation system to share knowledge.

Table 2: QCRI’s persona 2

6.2. Organizing the interface

We have organized our sitemap primarily based on our project descriptions and after meeting with our client we have clarified how our solution should look like. In addition, we have confirmed the user navigation as it shows in Figure 3. The admin and project will have a similar site excluding the login.

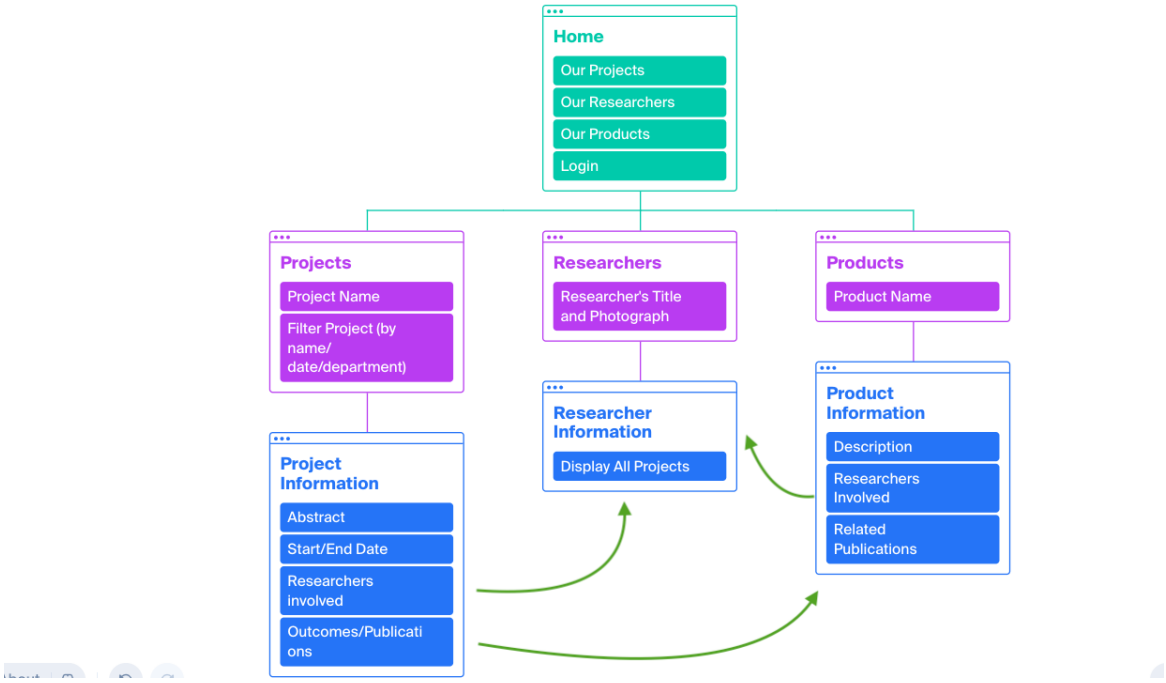


Figure 3: Sitemap for QCRI

6.3 Defining the standards

After consulting with the client, we have decided to use the existing Wiki.js page template. The design of the page is very simple and easy for the user to navigate. In addition, Wiki.js provides functionalities for customization based on the user preferences. However, when it comes to the page design, we will be using QCRI's logo as shown below.

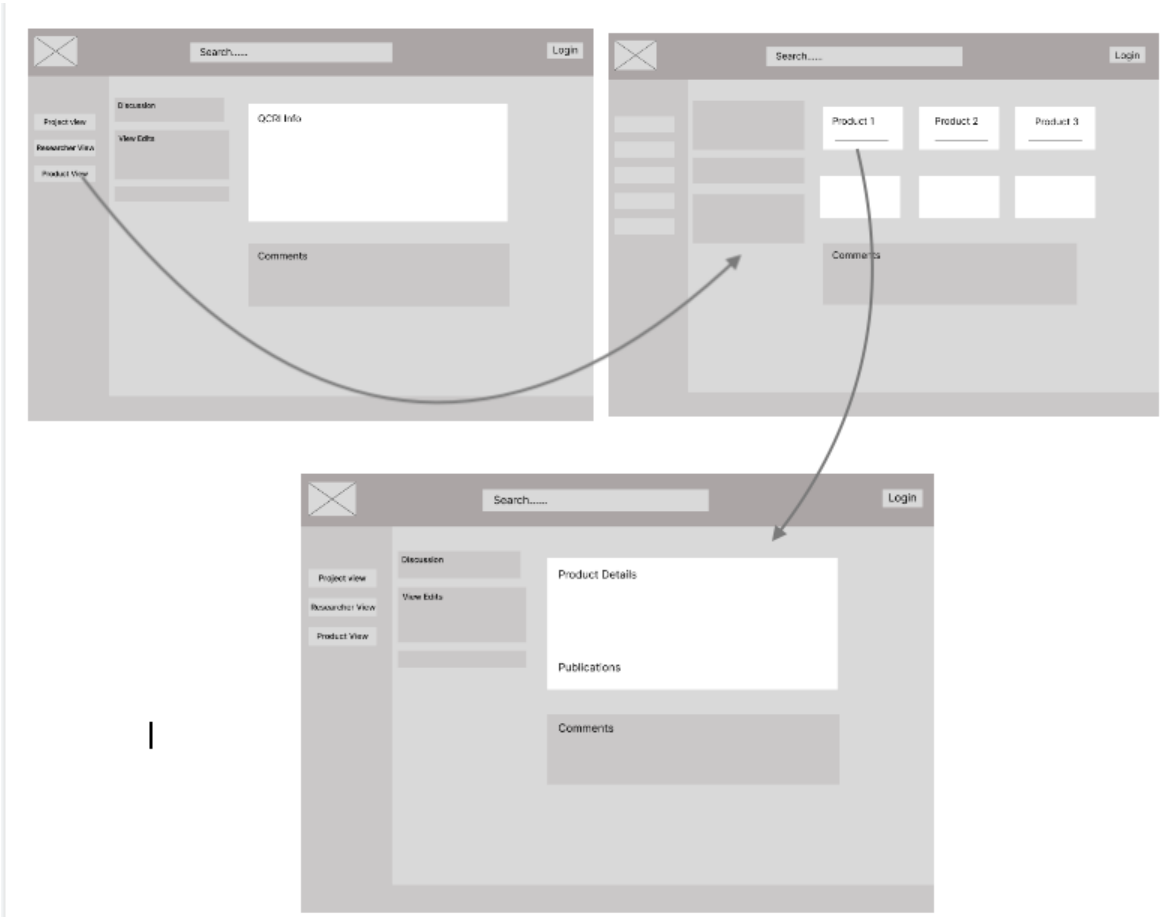
Figure 4: QCRI's logo

As for the color palette of the website, we have agreed with the client that we will abide by the same colors of QCRI's logo.

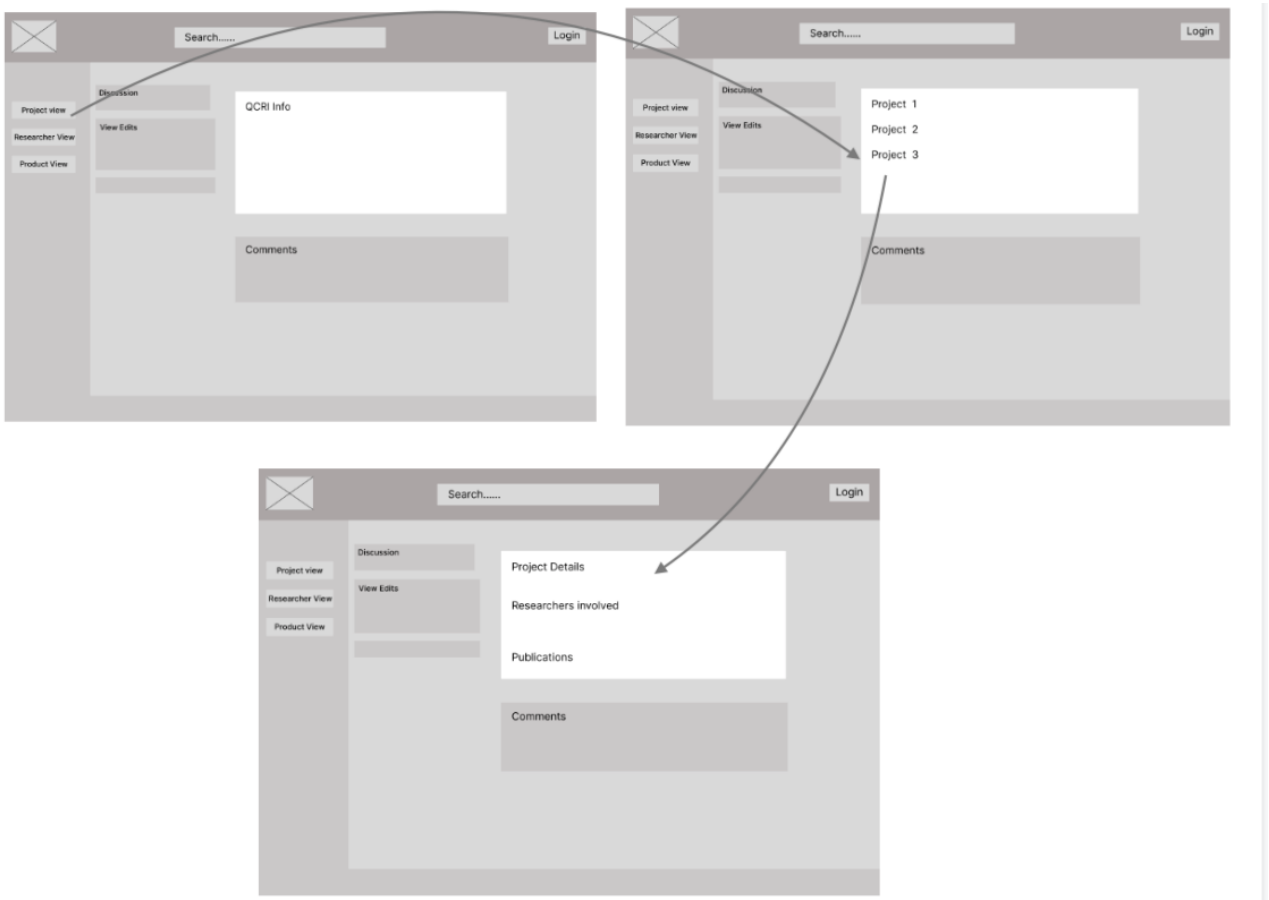
Figure 5: Color palette

6.4 Developing Prototypes/a Clickable Mockup

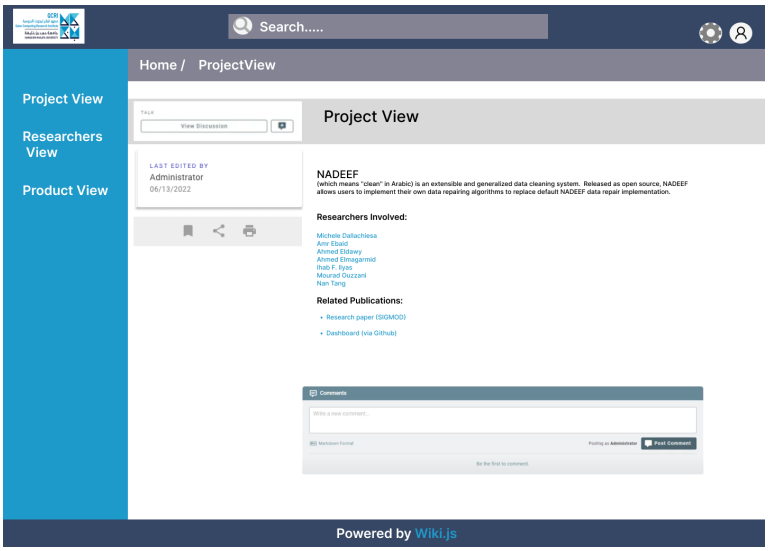
WireFlow 1:

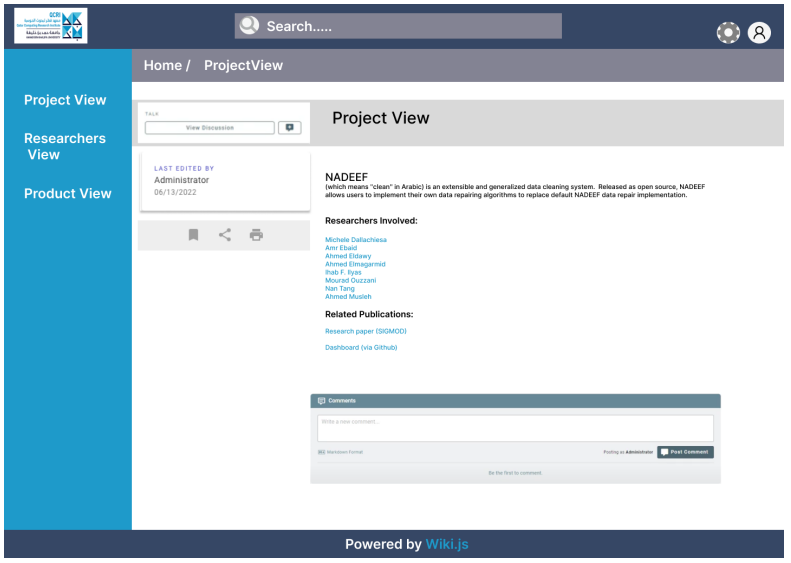

Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
QCRI Info
Comments
Search.....
Login
Product 1
Product 2
Product 3
Comments
Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
Product Details
Publications
Comments

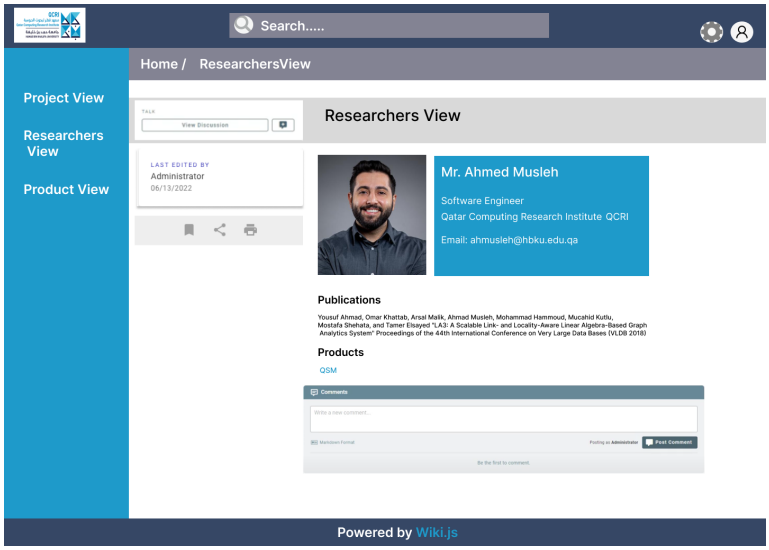
WireFlow 2:

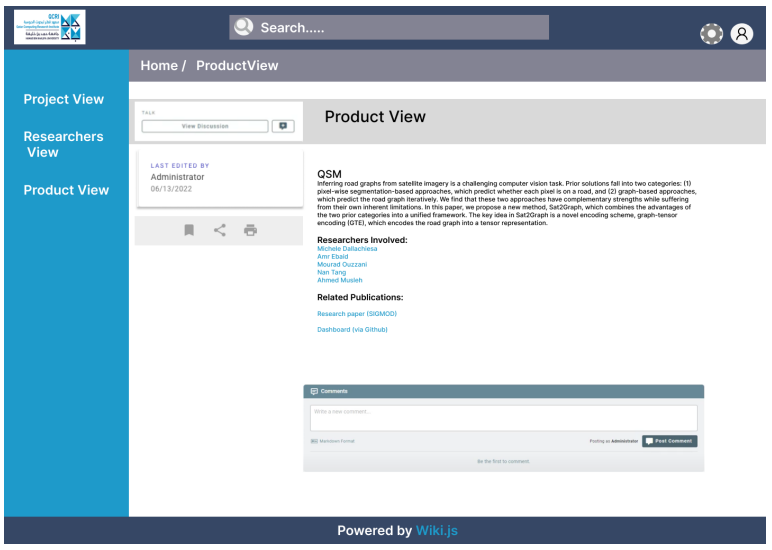

Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
QCRI Info
Comments
Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
Project 1
Project 2
Project 3
Comments
Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
Project Details
Researchers involved
Publications
Comments

High-fidelity clickable prototype:


Search.....
Home / ProjectView
Project View
Researchers View
Product View
Project View
View Discussion
LAST EDITED BY
Administrator
06/13/2022
NADEEF
(which means "clean" in Arabic) is an extensible and generalized data cleaning system. Released as open source, NADEEF allows users to implement their own data repairing algorithms to replace default NADEEF data repair implementation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Ahmed Eldawy
Ahmed Elmagarmid
Ihab F. Ilyas
Mourad Ouzzani
Nan Tang
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js


Search.....
Home / ProjectView
Project View
Researchers View
Product View
Project View
View Discussion
LAST EDITED BY
Administrator
06/13/2022
NADEEF
(which means "clean" in Arabic) is an extensible and generalized data cleaning system. Released as open source, NADEEF allows users to implement their own data repairing algorithms to replace default NADEEF data repair implementation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Ahmed Eldawy
Ahmed Elmagarmid
Ihab F. Ilyas
Mourad Ouzzani
Nan Tang
Ahmed Musleh
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js


Search.....
Home / ResearchersView
Project View
Researchers View
Product View
Researchers View
View Discussion
LAST EDITED BY
Administrator
06/13/2022
Mr. Ahmed Musleh
Software Engineer
Qatar Computing Research Institute QCRI
Email: ahmusleh@hbku.edu.qa
Publications
Yousuf Ahmad, Omar Khattab, Arsal Malik, Ahmed Musleh, Mohammad Hammoud, Mucahid Kutlu, Mostafa Shehata, and Tamer Elsayed "LA3: A Scalable Link- and Locality-Aware Linear Algebra-Based Graph Analytics System" Proceedings of the 44th International Conference on Very Large Data Bases (VLDB 2018)
Products
QSM
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js


Search.....
Home / ProductView
Project View
Researchers View
Product View
Product View
View Discussion
LAST EDITED BY
Administrator
06/13/2022
QSM
Inferring road graphs from satellite imagery is a challenging computer vision task. Prior solutions fall into two categories: (1) pixel-wise segmentation-based approaches, which predict whether each pixel is on a road, and (2) graph-based approaches, which predict the road graph iteratively. We find that these two approaches have complementary strengths while suffering from their own inherent limitations. In this paper, we propose a new method, Sat2Graph, which combines the advantages of the two prior categories into a unified framework. The key idea in Sat2Graph is a novel encoding scheme, graph-tensor encoding (GTE), which encodes the road graph into a tensor representation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Mourad Ouzzani
Nan Tang
Ahmed Musleh
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js

7. Implementation plans

7.1 Implementation and Integration of the solution components

The Wiki.js solution will provide a view for the projects that are being worked on, a view for the researchers' profiles, and a view for the project outcomes within this. Vue.js, a frontend JavaScript framework (Vue.js - the Progressive JavaScript Framework | Vue.js, 2014), JavaScript, HTML, and SCSS are used to build Wiki.js (requarks, 2022). The website's design will be done using HTML and SCSS, and Vue.js will be utilized to operate the website. Furthermore, the back-end database will be built via PostgreSQL and communication between the database and the interface will be facilitated with the REST protocol.

7.2 Development and Hosting environments

Our development environment will include Windows and MacOS, the two operating systems we'll be using, as well as Microsoft Visual Studio Code, version 1.71, which will be our IDE (Microsoft, 2021).

Git will be the version control tool we employ. The team as a whole has familiarity with Git, making it a rational choice. Git is also a dependable tool that will function well with the location where we intend to store our code, GitHub (Oluwatobi Sofela, 2020).

Our development system will be hosted in our local machines as Wiki.js does not require any additional software to be hosted. There is no need to host it on a cloud environment. All it needs is one computer for hosting.

8. Conclusion

To conclude, we will create a back-end database to store and manage QCRI projects, including publications, products, and researchers, and facilitate interaction with the database using an easy-to-use web interface. The database will use PostgreSQL, while the front-end web platform will use Wiki.js as the base and will be customized and developed to best fit the needs of our client.

9. References

requarks. (2022, September 4). GitHub - requarks/wiki: Wiki.js | A modern and powerful wiki app built on Node.js. GitHub. https://github.com/requarks/wiki

Client server architecture. Client Server Architecture - CIO Wiki. (n.d.). Retrieved September 11, 2022, from https://cio-wiki.org/wiki/Client_Server_Architecture

Microsoft. (2021, November 3). Visual Studio Code. Visualstudio.com; Microsoft. https://code.visualstudio.com/

Oluwatobi Sofela. (2020, August 19). Git vs GitHub – What is Version Control and How Does it Work? FreeCodeCamp.org; freeCodeCamp.org. https://www.freecodecamp.org/news/git-and-github-overview/

Vue.js - The Progressive JavaScript Framework | Vue.js. (2014). Vuejs.org. https://vuejs.org/

Faculty Biographies. Hamad Bin Khalifa University. (2022, March 11).from https://www.hbku.edu.qa/en/staff/mr-ahmad-musleh

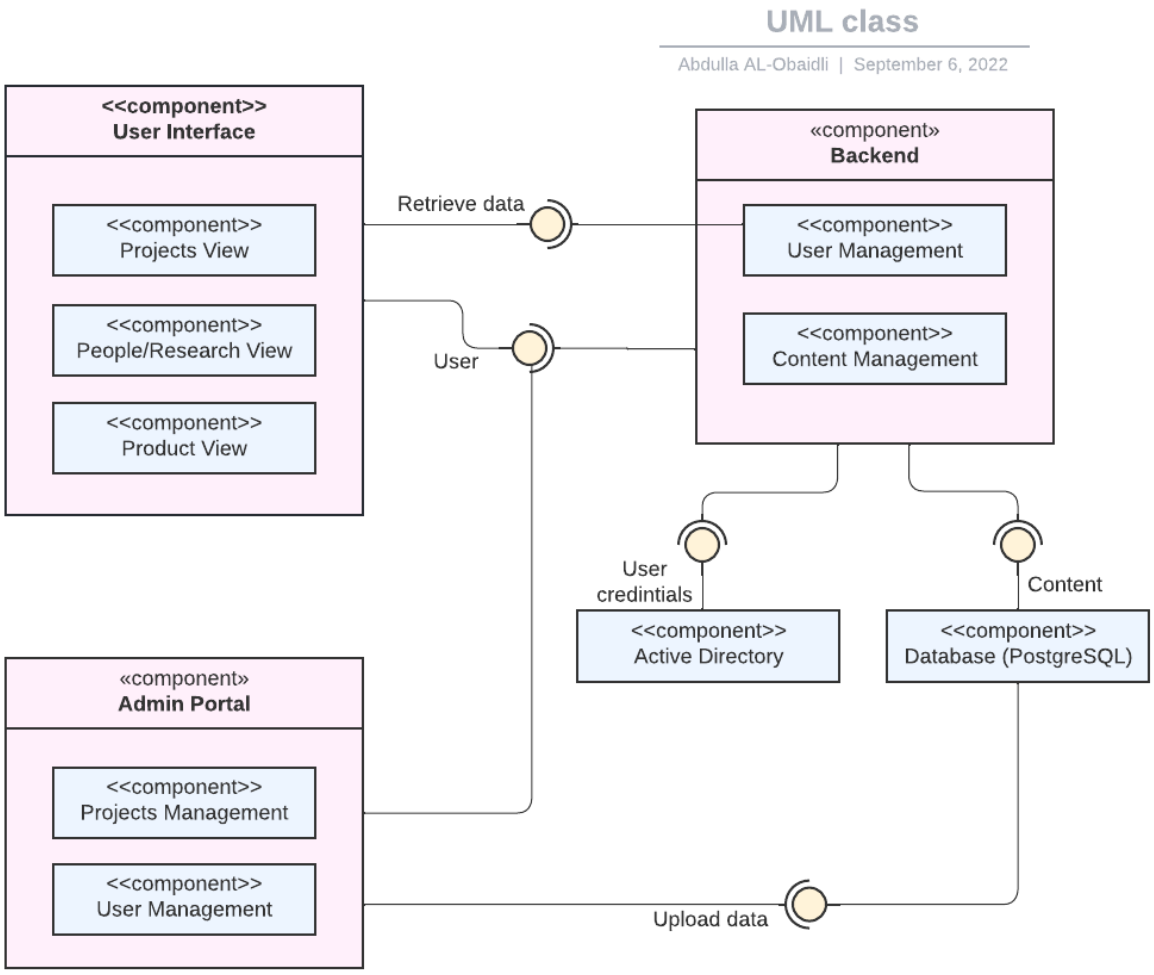
QCRI Projects and Demos | Hamad Bin Khalifa University. (2019, April 17). from https://www.hbku.edu.qa/en/qcri/da-projects-demos

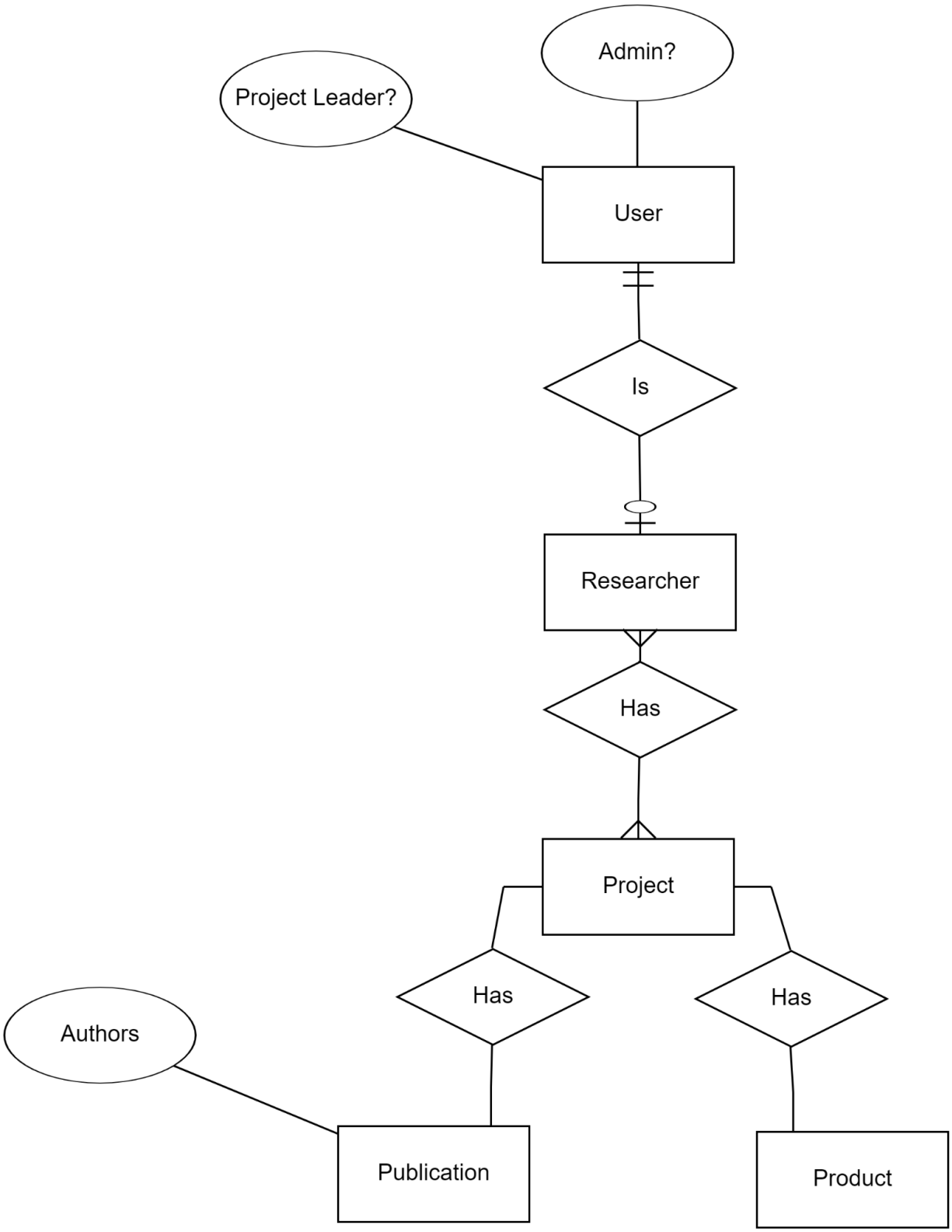
Products. QCRI. (2022) from https://products.qcri.org/

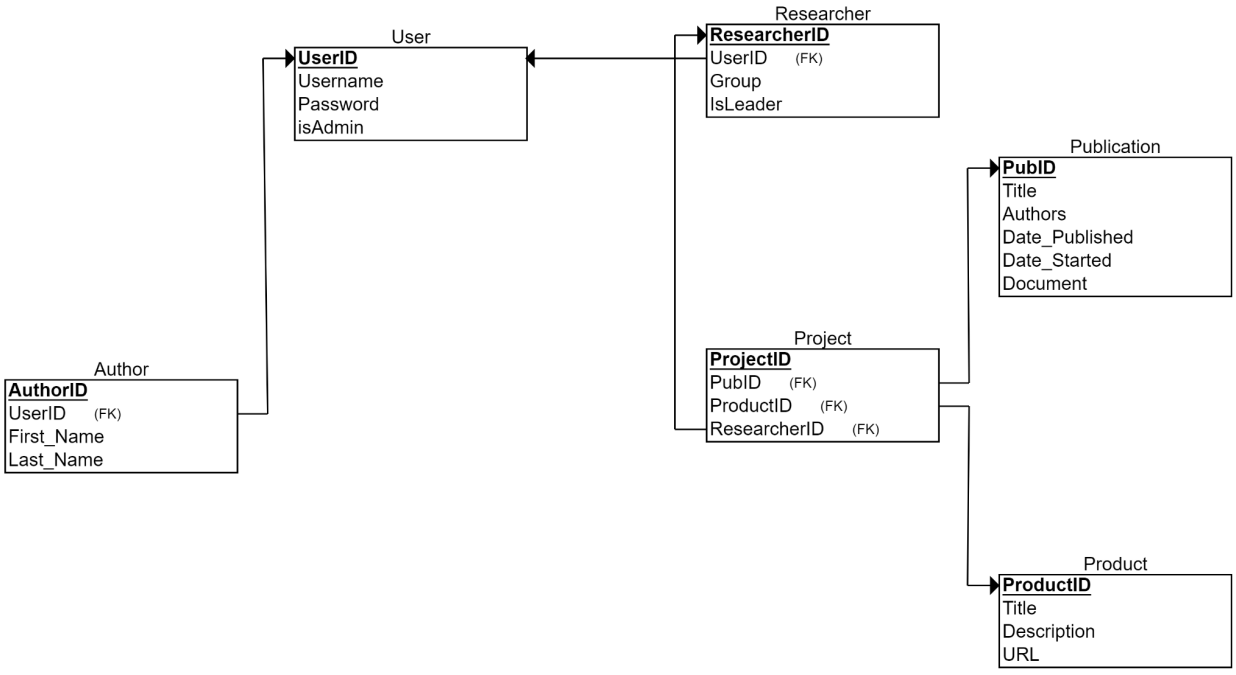
UML diagram types: Learn about all 14 types of UML diagrams. Creately Blog. (2022, September 9). Retrieved September 11, 2022, from https://creately.com/blog/diagrams/uml-diagram-types-examples/

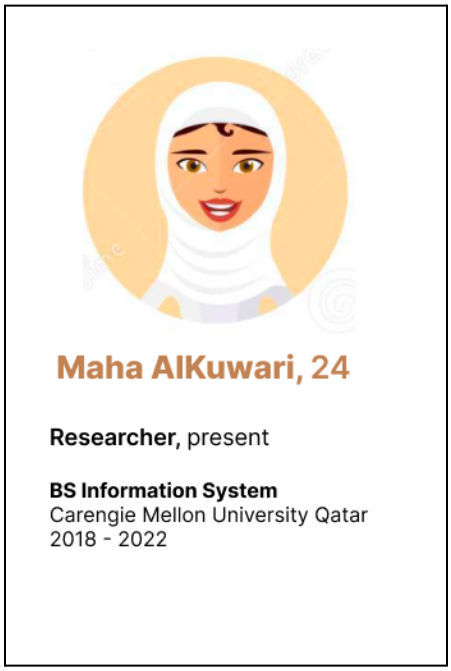
UML - Standard Diagrams. (2022). Tutorialspoint.com. https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm

10. Appendices




Admin?
Project Leader?
User
Is
Researcher
Has
Project
Has
Has
Authors
Publication
Product


User
UserID
Username
Password
isAdmin
Researcher
ResearcherID
UserID (FK)
Group
IsLeader
Publication
PubID
Title
Authors
Date_Published
Date_Started
Document
Author
AuthorID
UserID (FK)
First_Name
Last_Name
Project
ProjectID
PubID (FK)
ProductID (FK)
ResearcherID (FK)
Product
ProductID
Title
Description
URL

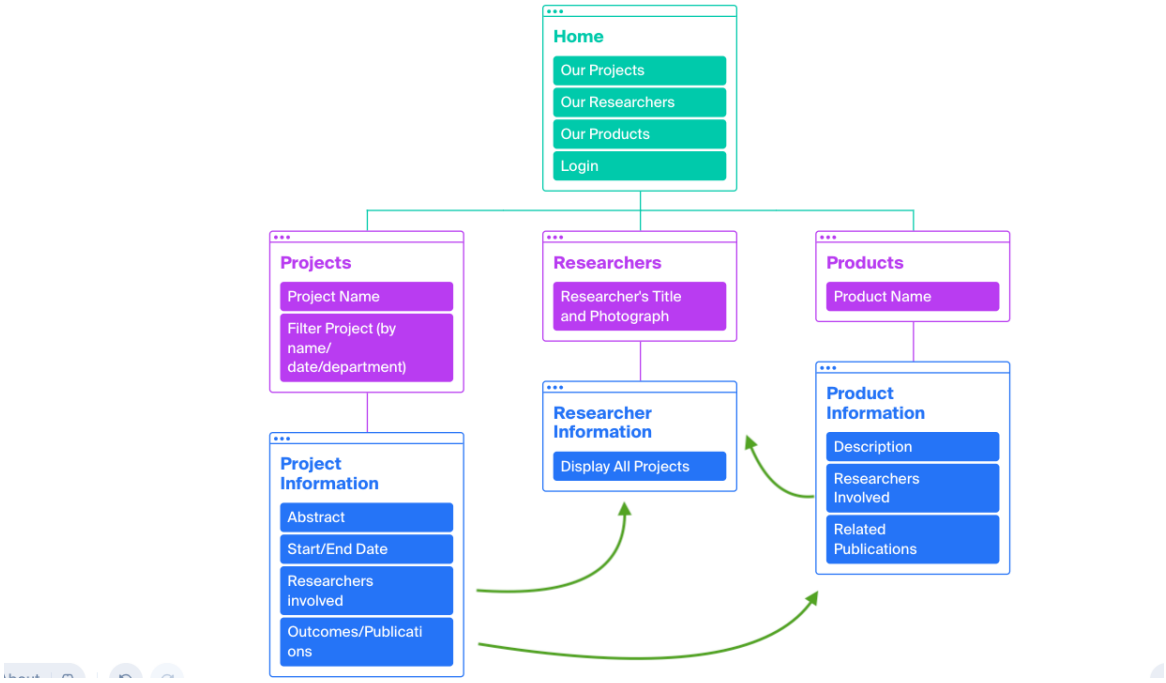

Maha AlKuwari, 24
Researcher, present
BS Information System
Carengie Mellon University Qatar
2018 - 2022

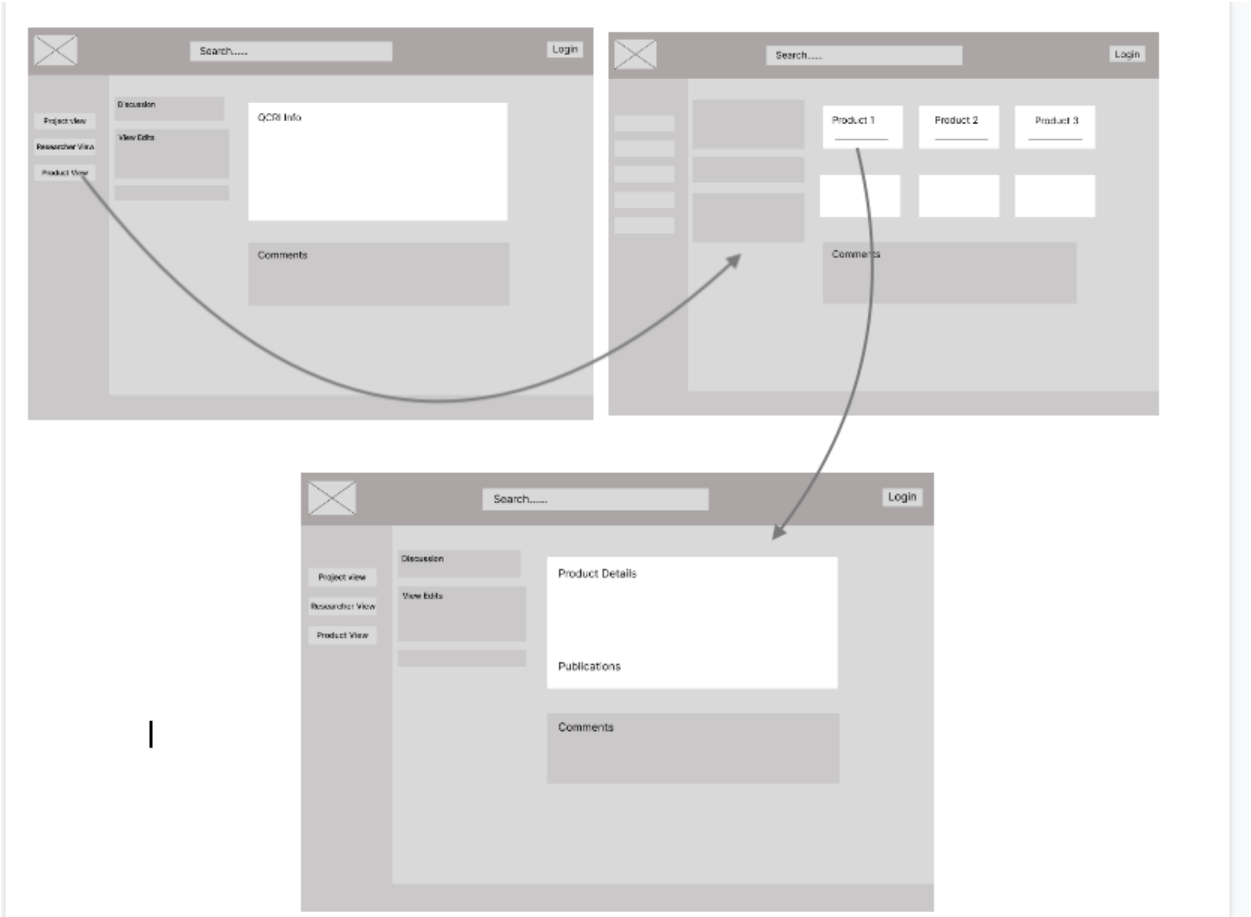
Mubarak AlSulaiti, 22

Project Manager, present

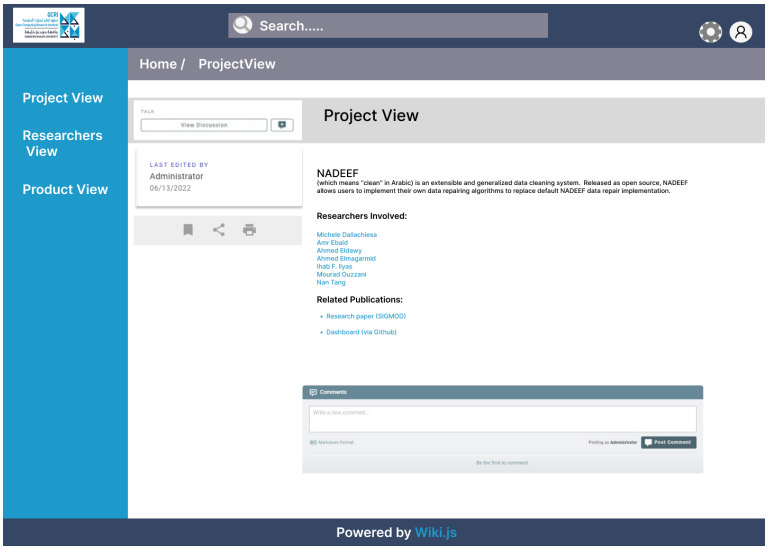
Research Asistant, 2017 - 2018

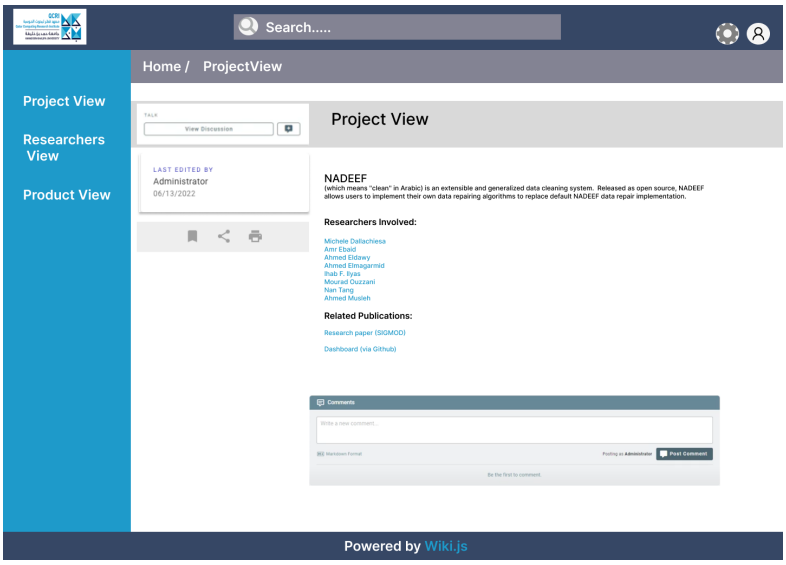
BSc. Computer Science
Carengie Mellon University Qatar
2011 - 2015

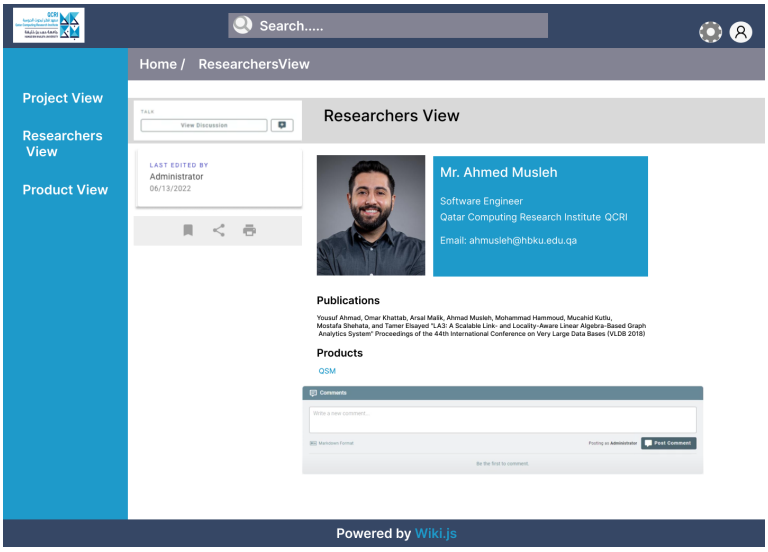


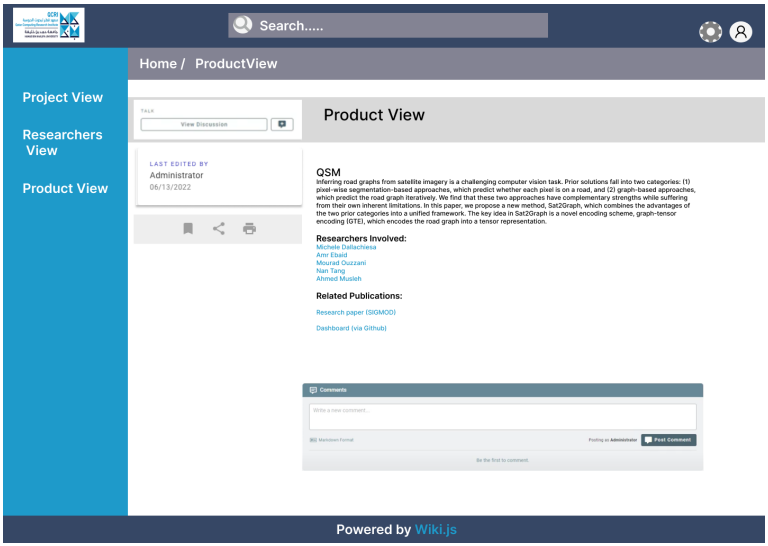

Search.....
Login
Discussion
View Edits
QCRI Info
Comments
Product 1
Product 2
Product 3
Product Details
Publications


Search.....
Login
Project view
Researcher View
Product View
Discussion
View Edits
QCRI Info
Comments
Project 1
Project 2
Project 3
Project Details
Researchers involved
Publications


Search.....
Home / ProjectView
Project View
Researchers View
Product View
Project View
View Discussion
LAST EDITED BY
Administrator
06/13/2022
NADEEF
(which means "clean" in Arabic) is an extensible and generalized data cleaning system. Released as open source, NADEEF allows users to implement their own data repairing algorithms to replace default NADEEF data repair implementation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Ahmed Eldawy
Ahmed Elmagarmid
Ihab F. Ilyas
Mourad Ouzzani
Nan Tang
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Post Comment
Powered by Wiki.js


Search.....
Home / ProjectView
Project View
Researchers View
Product View
View Discussion
Project View
LAST EDITED BY
Administrator
06/13/2022
NADEEF
(which means "clean" in Arabic) is an extensible and generalized data cleaning system. Released as open source, NADEEF allows users to implement their own data repairing algorithms to replace default NADEEF data repair implementation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Ahmed Eldawy
Ahmed Elmagarmid
Ihab F. Ilyas
Mourad Ouzzani
Nan Tang
Ahmed Musleh
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js


Search.....
Home / ResearchersView
Project View
Researchers View
Product View
View Discussion
Researchers View
LAST EDITED BY
Administrator
06/13/2022
Mr. Ahmed Musleh
Software Engineer
Qatar Computing Research Institute QCRI
Email: ahmusleh@hbku.edu.qa
Publications
Yousuf Ahmad, Omar Khattab, Arsal Malik, Ahmad Musleh, Mohammad Hammoud, Mucahid Kutlu, Mostafa Shehata, and Tamer Elsayed "LA3: A Scalable Link- and Locality-Aware Linear Algebra-Based Graph Analytics System" Proceedings of the 44th International Conference on Very Large Data Bases (VLDB 2018)
Products
QSM
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js


Search.....
Home / ProductView
Project View
Researchers View
Product View
View Discussion
Product View
LAST EDITED BY
Administrator
06/13/2022
QSM
Inferring road graphs from satellite imagery is a challenging computer vision task. Prior solutions fall into two categories: (1) pixel-wise segmentation-based approaches, which predict whether each pixel is on a road, and (2) graph-based approaches, which predict the road graph iteratively. We find that these two approaches have complementary strengths while suffering from their own inherent limitations. In this paper, we propose a new method, Sat2Graph, which combines the advantages of the two prior categories into a unified framework. The key idea in Sat2Graph is a novel encoding scheme, graph-tensor encoding (GTE), which encodes the road graph into a tensor representation.
Researchers Involved:
Michele Dallachiesa
Amr Ebaid
Mourad Ouzzani
Nan Tang
Ahmed Musleh
Related Publications:
Research paper (SIGMOD)
Dashboard (via Github)
Comments
Write a new comment...
Markdown Format
Posting as Administrator
Post Comment
Be the first to comment.
Powered by Wiki.js