Participatory Design: Data Model Management Application

CS4991 Capstone Report, 2023

Tillman Dean
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
gtd8fp@virginia.edu

ABSTRACT
A large McLean, Virginia-based financial company decided to streamline the complicated, manual process of adding new data models that wasted tons of man-hours. My team was tasked with creating an enterprise application to onboard new data models and view existing models. To accomplish this goal, we built a full-stack application using a combination of Python and Vue.js, that would be deployed on AWS and connected to a DynamoDB database. We spent an extensive time designing the application, involving all affected teams in the process. At the end of the summer, our application was deployed to a development environment and was able to perform the basic features we outlined. To bring the application into use, the database needs to be fully populated with the necessary data, and the application needs to be deployed through the pipeline into a production environment.

1. INTRODUCTION

When I was presented with the project last June, it seemed daunting – my team was tasked with reducing a several weeklong, man-hour intensive process into a five-minute self-service web application that any company employee would be able to use. We worked within the larger group that oversaw the company's enterprise document management platform. Every customer file or document was stored in this database. The process my team was tasked with streamlining dealt with the templates (types of documents) that the software stored. Often, when a certain area of the business was expanded, new types of documents with different types of data needed to be onboarded to the system.

The existing system involved the employee who needed the new template reaching out directly to a software team to manually create the data model and hammer out specifics, before uploading it to the database. To further complicate, the matter, there was no way to view the existing templates, so employees were less likely to suggest building off an existing template that could be similar to the new one that they needed, resulting in lost efficiency.

Additionally, the existing system used an outdated, relational RDS database, and the company had decided to switch to the NoSQL DynamoDB database. Our team stepped in to make an application that would allow employees to view existing templates and easily create and upload new templates to the DynamoDB database. Over the course of the summer, we used an agile process with a focus on participatory design to ensure our finished product best fit our stakeholders needs.

2. RELATED WORKS

Kautz (2011) explored an example of participatory design being used as a framework within the agile software development process. He found that engaging customers and users of the application early on and throughout the process led to significantly greater satisfaction with the product. My project made use of participatory design in a very similar manner, engaging with the teams involved in the current process to find common frustrations and issues before even beginning to code.

Hassan (2021) explores the advantages and disadvantages of relational and NoSQL databases, explaining why many, in the age of more and more diverse types of data, are making the transition away from relational databases. My project was a manifestation of some of Hassan's arguments about the disadvantages of relational databases, as the company had made the decision to migrate its data into a NoSQL database to improve scalability. When coding our new application we used a DynamoDB database hosted on AWS instead of the old RDS database.

3. PROCESS DESIGN

Our team was faced with a problem of making an application that was both intuitive to use and powerful enough to manage a very large and complex set of data. The only existing solution to view the data required was a simplistic and incomplete front-end application hooked up to the old RDS database. As a result, we had to scrap most of the backend code to connect to the new DynamoDB database and had to come up with the front-end for most of the application on our own.

We created a series of wireframe mockups on Figma for every page of our application before we coded anything. However, as interns, my team was unfamiliar with what a lot of the data meant and how the application was going to be used by company employees. To overcome this obstacle, we set up meetings with two different teams that would be using the software, speaking with both fellow software engineers and product managers. After showing them our mockups, we made several changes to the UI, including splitting the process of adding a new data model between multiple pages to make it feel more like a guided progression.

We coded up a new backend using Python and connected it to a new database temporarily populated only with dummy data. We then split up the pages for the front-end amongst our team and got to work making our application using the JavaScript framework Vue.js. In addition to adding the functionality to view, filter, and add data models to the database, we focused on making the application as usable as possible. One of the features I added to help accomplish this goal was a new drop down navbar that let the user easily navigate the application.

Another feature we added to make the process much easier for users was the ability to build from an existing model. Often models would be extremely similar but with a few key tweaks, or a model would need to be updated to a newer version or year. Building off an existing feature shaved the time required to input all of the data fields significantly, making the user much happier.

4. RESULTS

When completed, our application dramatically reduced the number of man-hours required to upload a new document data model to our database. The old process would often drag on for weeks, as it required several hours' worth of meetings between

different teams and required one team to hardcode the solution and manually add it to the database. Our new solution eliminated the need for these meetings and hours of hard coding, allowing the user to create a new model as quickly as ten minutes.

Our application also enabled the user to view the existing data models, a feature that did not exist before our application. This made using the models much easier and the entire system more transparent to everyone in the company who interacted with it.

5. CONCLUSION

This project proved to be incredibly valuable to the company, who were aiming to move as many applications to be self-service as possible. Our application removed the need for software teams to hand-make every new data model, replacing several hours of meetings and coding with only a few minutes that could be done by the employee who needed the model.

Our application also provided the first coherent way of viewing all of the existing models which allowed for greater understanding of the way in which they were used. By meticulously involving all of the stakeholders in the design process, we were able to create a product that was useful and intuitive. Over the course of the project, I gained valuable professional experience of both technical skills and soft skills like how the agile development process runs on a team. I also learned to greatly appreciate the design process and not rush into coding a solution right away.

6. FUTURE WORK

Our application was deployed into a development environment, but it still needed some work to be deployed into production. First, the database needs to be populated with the actual data models from the old RDS database, as we were still using dummy data in our DynamoDB database. Second, the deployment files for the company's internal pipeline need to be configured further to allow it to deploy further. Once these two steps are completed, the application will be able to be used by actual employees.

REFERENCES

Hassan, M. A. (2021). Relational and NoSQL Databases: The Appropriate Database Model Choice. *2021 22nd International Arab Conference on Information Technology (Acit)*, 705–710. https://doi.org/10.1109/ACIT53391.2021.9677042

Kautz, K. (2011). Investigating the design process: Participatory design in agile software development. *Information Technology & People*, *24*(3), 217–235. https://doi.org/10.1108/09593841111158356