



DACST Database Requirements

CS 330

Database Design, Development and Deployment

Fall 2021

Version 4.0

REVISION HISTORY			
DATE	VERSION	DESCRIPTION	AUTHOR
10-25-2021	2	Updated user overview, hardware interfaces and requirements, and software interfaces	Makaylah / Erik
10-27-2021	2	Add Features	Makaylah / Erik
11-15-2021	3	Updated to include website details	Makaylah

Table of Contents

PROJECT NAME	1
Version 4.0	1
INTRODUCTION	3
PURPOSE	3
INTENDED AUDIENCE AND PERTINENT SECTIONS	3
PROJECT SCOPE	3
REFERENCES	3
DESCRIPTION	4
TEAM PRODUCT PERSPECTIVE	4
TEAM PRODUCT FEATURES	4
USER OVERVIEW	4
OPERATING ENVIRONMENT	4
CONSTRAINTS: IMPLEMENTATION / DESIGN	4
DOCUMENTATION	4
ASSUMPTIONS / DEPENDENCIES	6
SYSTEM FEATURES	6
SYSTEM FEATURE 1: Data Storage	6
SYSTEM FEATURE 2: Querying database by tags and keyword search	6
SYSTEM FEATURE 3: Download Data as .CSV files	7
REQUIREMENTS OF EXTERNAL INTERFACE	7
USER INTERFACES	7
HARDWARE INTERFACES AND REQUIREMENTS	7
SOFTWARE INTERFACES	7
COMMUNICATION INTERFACES	8
APPENDICES	8
APPENDIX A: GLOSSARY OF TERMS	8
APPENDIX B: ANALYSIS DOCUMENTATION	8
APPENDIX C: ISSUES	8
BACKGROUND DETAILS	9
PROTOTYPE DIAGRAM	9
BACKGROUND RESEARCH SUMMARY	9
TEAM MEMBER RESPONSIBILITIES AND BENCHMARKS	9
DATA DESIGN	10

INTRODUCTION

PURPOSE

Identify and describe scope of product whose technical specifications are being documented, and describe desired outcome.

The purpose of this project is to understand students' perception of Computer Scientists. The information being stored is text and images collected from students. Organizing this data in a well defined database will be useful for further analysis of this existing data as well as for storing new data in the same database.

INTENDED AUDIENCE AND PERTINENT SECTIONS

Describe each type of reader, including developers, users, testers, writers, marketing, etc., and outline which portions of the document are crucial to their department.

The developers of this database will use the "system features" and "requirements of external interface" to design the database. The users are people who will designate the requirements of the external interface. Testers will need to understand the purpose of the project to ensure that the database is useful.

PROJECT SCOPE

Describe relevant benefits, objectives, and goals.

The objective of this project is to create a database that collects images and related text for analysis on how people perceive computer science/scientists. This database will facilitate the classification of images by creating human-labelled training sets for machine learning.

REFERENCES

List any referenced document names or links.

Manzo Harmon Doore ACM SIGCSE poster 2019 022819.pdf
<https://dl.acm.org/doi/pdf/10.1145/1041624.1041628>

DESCRIPTION

TEAM PRODUCT PERSPECTIVE

Provide context and origin of team product, along with expected functionality.

The product is being designed to host the data for a form of the "Draw-A-Scientist" Test (DAST) which is used to assess perceptions of scientists. It will be used to store the drawings and responses of surveyed individuals, allowing ease of access in order to better understand perceptions and misconceptions of the computer science field. The origin of this study is a paper by C.D. Martin in 2004 where she identified the harm in perceiving computer science as "hardware, software, and programming".

TEAM PRODUCT FEATURES

List main features with brief description.

The main features of a database created for this product are the ability to store and access text information as well as submitted drawings. The added benefits of storing both types of information is that once they are organized they can be used in machine learning to train an AI to label the drawings. This metadata will give the opportunity to perform statistical analysis on "the drawings".

USER OVERVIEW

Define groups, and describe user characteristics.

The users are researchers who will be able to query the data by tags or keywords. These users may not be programmers and do not need to know any back end information. As a result, it is important to the design of this database that these users can easily add tags to existing documents. Researchers may also need to add new data samples to the database. To facilitate the use of data outside of the database it is important that the data can be downloaded as a ".csv" file.

Another user is a student doing data entry. Though document insertion is streamlined through google forms the student still needs the capabilities to add tags to documents.

OPERATING ENVIRONMENT

Illustrate where software (or hardware) will function (e.g., operating systems, languages, cloud platforms, and other components or applications) or which it must work in conjunction (e.g., other team products).

We plan to create a document-oriented database. We plan to use MongoDB to manage this database. Since user's of the database will be a mix of skilled programmers and students the UI is important for increasing the accessibility of the database. We will use a website created through node.js, html, and css.

CONSTRAINTS: IMPLEMENTATION / DESIGN

Describe limitations impacting development. Note: time constraints are a given, but what else presented barriers to the design and development?

We are using unfamiliar software. This is also the first database that we are being asked to design. We don't have a background in website design. The commands for getting into the database aren't easily scripted.

DOCUMENTATION

Describe content, mode of delivery, and standards.

The content of this project is a database that can be used to easily store and access data associated with the DACST. Delivery will be simple because the class submission will be the same as delivering to our client. We have a standard of professionalism.

ASSUMPTIONS / DEPENDENCIES

Detail all assumed factors that could potentially impact the product's future implementation.

One possible change could be if there is a need for a cloud based database. We may also need to collect similar information for other studies. Some assumptions that we have are that the software we are using is free/open source and that there is not a priority on data security.

SYSTEM FEATURES

SYSTEM FEATURE 1: *Data Storage*

DESCRIPTION AND PRIORITY	The MongoDB database will hold text and image data, with associated tags for each response. This feature is of high priority as other features can only be created subsequently.
STIMULUS / RESPONSE SEQUENCES	The stimulus for this feature is the input responses, and the response is the populated database.
FUNCTIONAL REQUIREMENTS	MongoDB allows for the insertion, update, and deletion of responses in the database.

SYSTEM FEATURE 2: *Querying database by tags and keyword search*

DESCRIPTION AND PRIORITY	This feature will give the user the ability to search through the data for the desired information based on tags and keywords that are populated through the input data. This feature is dependent on database creation, thus it is of lower priority than the database.
STIMULUS / RESPONSE SEQUENCES	The stimulus is searching by tags for keywords, and the response is the returned search results.
FUNCTIONAL REQUIREMENTS	<p>The tags will be stored as an attribute.</p> <p>This source demonstrates how to tag and search by tags using a java dependency. This includes all the functionality required of this system feature.</p> <p>https://www.baeldung.com/mongodb-tagging</p>

SYSTEM FEATURE 3: *Import / Export Data of .CSV files*

DESCRIPTION AND PRIORITY	<p>The user should be able to select data on a large scale and retrieve the information in a .CSV file.</p> <p>This feature is important for analyzing stored data. This makes the feature a high priority since it will enable researchers to script a neural network to automatically classify images based off of high quality researcher-labelled tags.</p>
STIMULUS / RESPONSE SEQUENCES	<p>Stimulus: User selects data to be exported as .CSV</p> <p>Response: Bundle data into a new .CSV file and return to the user.</p>
FUNCTIONAL REQUIREMENTS	<p>This Feature depends on the tagging system (feature 2).</p> <ul style="list-style-type: none">- "mongoexport is a command-line tool that produces a JSON or CSV export of data stored in a MongoDB instance"- The MongoDB compass UI also has an easy export feature.

REQUIREMENTS OF EXTERNAL INTERFACE

USER INTERFACES

Describe product / user interface characteristics, including standards, style guides, constraints, functionality, and sample screens if applicable.

The UI will be a website. The UI will be simple allowing users to search and download specific queries based on tags or the full dataset. Entered text data is expected to be spelled correctly and capitalization will be ignored in later analysis of the data. The website will have a link to a form which will allow users to upload their information for the DACST. The website will also have a page for adding tags to documents.

HARDWARE INTERFACES AND REQUIREMENTS

Describe product / hardware interface characteristics, nature of interactions, and communication protocols.

The data will be stored in a MongoDB database. The user can access the website on any internet capable device to access the database with ease. Data will be stored on MongoDB Atlas for decentralized cloud storage, allowing the user to access the database from any device. Aside from querying and manipulating the database, users can also download data as a ".csv" file.

SOFTWARE INTERFACES

Describe product / other software interface characteristics, including component names and versions, databases, operating systems, libraries, tools, etc. Specify any constraints, along with the nature of communications and what data is coming in and being disseminated.

The UI will be a NodeJS website with express. The pages will have a navigation menu, a carousel of DACST images, and a query form that features check boxes, a search bar, and buttons. The page will have links to outside pages(google forms and google image links). We want to make sure that our href links open in a new tab, to prevent the user from not being able to return to the DACST page, but also

make sure that there is no redirect on the main page during the link open (done with the "rel="noopener noreferrer" attribute). This is especially important for the "Take the DACST" menu item which links to a google form that can automatically populate MongoDB.

COMMUNICATION INTERFACES

List requirements of communication functions: email, browsers, servers, forms, etc. Describe standards employed, security or encryption measures, data transfer rates, and synching.

We have a main form for querying the database. Once relevant information has been pulled from the database, the user can then download that data into a .csv file. This is an important standard because it will make it easy to use this data for AI classification:

Data should be able to train an AI. Once a classifier has been created, a script can pull newly submitted drawings, classify them with the AI, and then update the processed image data for each new response. This in conjunction with the connected google form will allow for the automated collection of student data in an anonymous and consistent way.

APPENDICES

APPENDIX A: GLOSSARY OF TERMS

Define all terms and unique acronyms employed throughout the document and specific to the project.

DAST - Draw a scientist test

DACST- Draw a computer scientist test

UI - User interface

GUI - Graphical user interface

Database Fields:

APPENDIX B: ANALYSIS DOCUMENTATION

List file / document names / provided links to all diagrams, models, additional findings pertinent to technical specification development.

DACST Study Protocol 082819 Final.docx

DACST protocol Section A

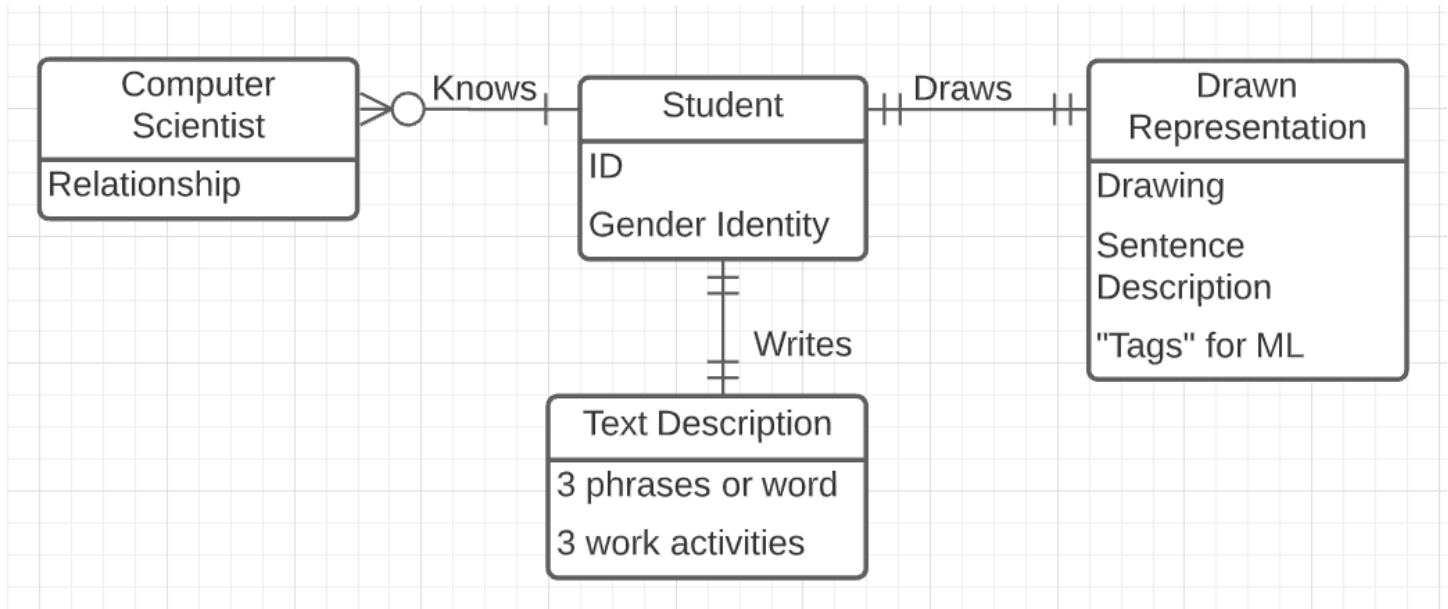
APPENDIX C: ISSUES

List all unresolved issues, TBDs, pending decisions, findings required, conflicts, etc.

ISSUES		
ID	DESCRIPTION	PARTY RESPONSIBLE
1	Find a way to make it easy to upload to the database (via google forms?)	Makaylah/Erik
2	Website Accessibility	Erik
3	Fix download	Erik
4	Access data in app.js /getdata	Makaylah
5	Website Aesthetics	Erik/Makaylah

Background Details

Prototype ER Diagram



Background Research Summary

We have been researching MongoDB and the tools that can be used with it. We chose MongoDB because of its document based database structure which is necessary because our data will contain mixed media (text and images) and may contain nulls which are not as easily supported in relational databases. MongoDB has an accompanying GUI, MongoDB compass, that allows for easy analysis of documents. This will be useful for monitoring the status of the database and creating views for the UI. To download data, we plan to use mongoexport which will allow for users to download data into .csv and .json files.

Team Member Responsibilities and Benchmarks

Task Name	Description	Member(s) responsible	Due Date
Collect Model Data	Get sample data from client / generate samples independently	Makaylah	11-5-2021
Set Up Document DB	Download MongoDB/ compass	Makaylah/Erik	11-5-2021
Insert Model Data	Add Data to existing MongoDB document DB	Erik	11-12-2021
Add Tag and Search Functionality	Tags will be an embedded attribute	Makaylah	11-24-2021
Create Views	Add client required views (must collect view requirements)	Makaylah/Erik	11-24-2021

Test Data export	Verify the generation of output files	Erik	11-24-2021
------------------	---------------------------------------	------	------------

Data Design

Key Value Pairs:

___*STUDENT*___

CODE (student_id)

GENDER

SEMESTER

CLASS

ADMINISTRATION

___*IMAGE*___

MARTIN SCORE

IMAGE LINK

IMAGE GENDER

POSITIVE IMAGE

PICTURE DESCRIPTION

EMBEDDING 1 - FEW:

TAGS

THREE WORDS/PHRASES

COMPUTER SCIENTIST ACTIVITIES

REFERENCING many/ infrequent data:

CS_ID

CS_ID

KNOWN COMPUTER SCIENTIST

CS INFO