Team 5 (Delta V Innovation Database Team)

Database Support Requirement Specifications

9/17/18

Authors:

Austin Rice:

-Development and Target Environments

-Function Requirements

Matthew Stipsits:

-System Model

-User Interaction

-Feasibility

Daniel Palmer:

-Introduction

-Project Overview

-NonFunctional Requirements

Customer:

Mike Flamm

deltaVinnovationsinc@gmail.com

Table of Contents

Introduction

The overall goal of this senior design project is to work with two other groups on modifying an existing software specializing in Computer Aided Design for crash and crime scene recreation, storage of captured data, and sharing amongst users: data, analysis and simulation of events. The area of development that Team 5 is working on is the improvement of the database that has been created by last years senior design team. The main features to be added this semester include: populating a considerable amount of data into the existing database, creating more tables in the database, allowing for user's to add information into the database, and creating a login page for users. The database will take input from the mobile application and will allow users to share it with the desktop application or other users. As for who the users are, Delta V Innovations intends to offer this software application to numerous users including but not limited to insurance agencies, private investigators, and car manufacturers. More details on the user interaction and scope will be described in the project overview section of this report. Information on the hardware and software resources necessary to build and run the system will be described in the development and target environments section. The system models section will present a high-level view showing the major components of the existing and proposed system. The user interaction section incorporates use-case diagrams and use case scenarios to explain how the user will be interacting with the application. The functional requirements section goes into depth on the changes the team will be making with the databases, along with the function of the user being allowed to add data. Then, the nonfunctional requirements section will talk about the database compatibility, size constraints, and efficiency. Finally, the feasibility section will talk about

whether the goals set out to complete are realistic, and whether or not completing them will be a problem.

Project Overview

In order to successfully complete the goals brought forth to the group, an in depth analysis on the project scope, requirements, and constraints must be looked at. As mentioned in the introduction section, the users of the application and database that groups are working on include insurance agencies, private investigators, and car manufacturers. This is not a definitive list of users; the list will only grow the more we enhance the database and application. A general overview of the application is that it takes in conditions such as traffic flow, time of day, tire marks, weather conditions, etc.. This system seeks to help authorities determine where and why accidents are occurring in the hope of preventing the likelihood of them occurring. The database side of the project requires that user's will be able to store vehicle information as well as query vehicle information. The previous database team was able to create the database, as well as populate it with a good amount of data. They were able to create the required tables and fields for initial use, but this years database team is required to populate the database with more tables for incoming data. The previous group also established a link between the application and the database, so that the two could communicate with each other, but not send information. What has to be done this semester is a general maintenance of the database to ensure efficiency, an enhanced communication between the application and the database so that information can be effectively transferred, and an import of lots of vehicle information. With these features added, the database will be ready to take in vehicle data from users, query information, and return any

information needed from the users. There are two main constraints that the group will be dealing with the entire semester, and those constraints are the efficiency and storage/size of the database. Companies do not want a database that is too slow to retrieve or send information, and the larger a database gets, the less efficient it gets. So trying to balance the two, and increase efficiency through code is going to be very important.

Development and Target Environments

There are a few hardware and software resources that are required in order to support the database for the mobile app and in order to run the application itself. The main hardware and software resources that will be used in the maintenance of the database will be AWS, more specifically the RDS portion of AWS, and SQL. These tools will be used to query the current tables and add more tables to the preexisting database. There are not many resources that are required by the end user to run the application besides having a device with access to it.

The database will be located within Amazon AWS. Amazon Web Services is hosted by Amazon and provides a platform for cloud computing. Last year the former team that worked on this project decided to go with AWS due to its plentiful documentation on how to use it and the fact that it offered extra security features that other providers did not. The specific portion of AWS that will be utilized is RDS which stands for Relational Database Service. This is where the database that is accessed by the mobile app will live. Additionally, the database engine that the RDS will be utilizing will be MySQL. SQL queries will be written in order to add data to the database and allow the mobile app access to the data. This will most likely include select and insert commands; however more commands may be necessary.

The end users will be accessing the application through either Android or iOS. It is available on the app store for download and is free to use. Even though the database is a major part of the application the user will have no direct interaction with it. Instead the UI will direct the user to either conduct a search on the database or add data to it. In order to test the app the team will be downloading the app the same way an end user would. This allows for proper testing of the software in its natural environment. In summary, the only requirement for running the mobile application is access to either an Android or iOS system and internet.

System Model

In order to complete our current goals we need to understand what has already been done on our project.

Currently, the database for Delta V Innovations only contains a small amount of car information to be used in calculations. It already has the ability to be called upon from both the website and the mobile app. The tables are organized neatly by year, make, model, and different versions. Our proposed system will include a vast amount of information on all cars from 1970 to present as well the ability for users to store data from via the website or the mobile app. We will be utilizing the organization of the tables already within the database to incorporate th

User Interaction

Among the features Team 5 will be implementing this semester is the user login page as well as allowing users to add and store information into the database. Team 5 will develop a way to validate user credentials and keep a table of users with access to the database. After validation

is complete, the user will have access to the rest of the data within the database as well as access to the ability to store and access their own personal data.

The two main user interactions that we will be implementing is storing data and retrieving data. For adding data, our system must first verify the user has rights to store data. After verification is completed, the user will be prompted to enter information and click the store button. From there the system will store the entered information into the database and then validate that the information has been stored. Once the validation is complete the user will be shown a message confirming the information has been stored.

For retrieving data, the user will search through the database by entering information to search for and clicking the search button. The system will then query the database, conduct calculations if necessary, and then returns the results to the user. Finally, the user will be presented the results of his search.

Functional Requirements

There are four main functional requirements that we plan to accomplish this semester. These include importing data into the existing database, adding new tables, writing queries to allow the application access to data, and creating a way for a user to add personal information that will be secured. Each of the requirements will now be outlined in more detail and more information will be provided in order to allow verification and testing of each.

The first requirement that will be completed will be adding more vehicle data to a database table that was created last spring. The customer has made this task highest priority and therefore it will be accomplished first. Currently all of the data that needs to be stored in the

AWS database is contained within many CSV files. In order to retrieve this data and import this data in an efficient manner a script will be created. Before creating this script research will first be done to make sure that a similar script does not already exist. The estimated number of story points for this user story is 20 points and the estimated time that it will take to finish is roughly 2-3 weeks. Once complete it will be easy to verify the completion of the story by simply combing through the database and making sure that all of the CSV data made its way in. It will be more difficult to test whether this story was done properly. Testing will be conducted by using the mobile application and verifying car information that wasn't available before is now available.

The user story that has been emphasized by the customer to be second priority is the addition of new database tables that align with what the mobile teams are able to accomplish. The tables that need to be added down the road consist of cases which are comprised of vehicle information, occupant information and users. In order to come up with the structure of the database tables communication must occur between the mobile teams and the database team. One benefit of using an AWS database is that there is a lot of documentation on how to interact with the database. Therefore, once the tables that need to be added are structured it should be relatively easy to add new tables and use the script from user story one to import the data. This user story will be verified by confirming that the new database tables match up with the provided ER diagram and relationship schema. There will not be testing for this user story since there is no way for the mobile app to access this data. Instead this story will be tested once user story three is complete.

In order to access the data that was added in user story three SQL queries will need to be written. This functional requirement is of very high priority since it is essential in allowing the user to interact with the database. Most likely this story will be done in collaboration with one of the mobile teams since the code will be written in the codebase they have created. The SQL queries themselves should not take long to create; however it is essential that performance testing is completed on the queries. Many times a database call can cause the whole application to drastically slow down. This story will be verified by confirming that the data is accessible within the mobile app. In addition to performance testing, manual testing will be done using the app in order to verify that the correct data is being pulled from the database.

The last big functional requirement for the semester is allowing the user to enter personal information into the database. This information will need to be secured in a way that no other users can see it. At this point in the planning process the customer has specified this user story as a stretch goal and therefore it is of lower priority than the other stories. In order to accomplish this task a login page will need to be created by the mobile development team. This way the data entered can be tied to a specific user within the database. If accomplished this user story will be verified by confirming that users are only allowed access to the data that they enter into the database. Testing for this story will be extensive due to the new functionality added to the app and the massive number of edge cases that will exist.

Nonfunctional Requirements

It is important to understand the nonfunctional requirements of the project before it is started. These requirements usually detail the constraints and the limitations to the project. For

the database, the biggest and probably most important nonfunctional requirement is query or processing times. It is crucial that the data presented and retrieved is done so in a timely manner, and doesn't take too much time to accomplish. Especially as the database size increases, we do not want code or data structures that cause response time to triple in time.

Next would be database size constraints. The team is in charge of inputting a large amount of data, and create tables so that more data will be ready to be inputted into the database. With this being said, the database team needs to be sure that the inputted data meets the database size constraints and if it doesn't, then thought needs to go into how much space is needed.

Another nonfunctional requirement would be the ability to handle bad data and ensuring data integrity. We want to make sure that the database does not break when bad data is inputed, and that the application itself will handle the situation. We also want to ensure that the data inputted is accurate, consistent, and traceable. The relationship schema and ER diagram should be able to accurately describe the storage and connections of all the entities and attributes within the database, allowing everything within the database to be recoverable and searchable.

The last nonfunctional requirement is that we need to ensure the security of the users that make accounts on this application. Whenever users are given the opportunity to login or purchase a subscription to something, security becomes an issue. It would be beneficial to ensure passwords fit requirements (length, special characters) that maximize security. Also, inactivity logouts are needed just in case the user forgets to logout. Access levels are also needed to give some users elevated privilege, and to make sure others don’t get certain privileges. For this application, a non-subscribed user should not have the ability to input vehicle information into the database, but subscribed users should have this ability.

Feasibility

Although we have many ideas for the project's direction and as much as we have planned for it, we still need to take a step back and look at the overall feasibility of our goals. Our main focuses to look at are whether or not we can even complete this project on time as well as looking at the minimum and maximum goals we hope to accomplish.

The three of us have the advantage of being roommates so we are always within reach for project discussion and coordination. With our workforce, schedule, and close proximity of team members completing the project and reaching our desired goals is definitely possible and within time constraints.

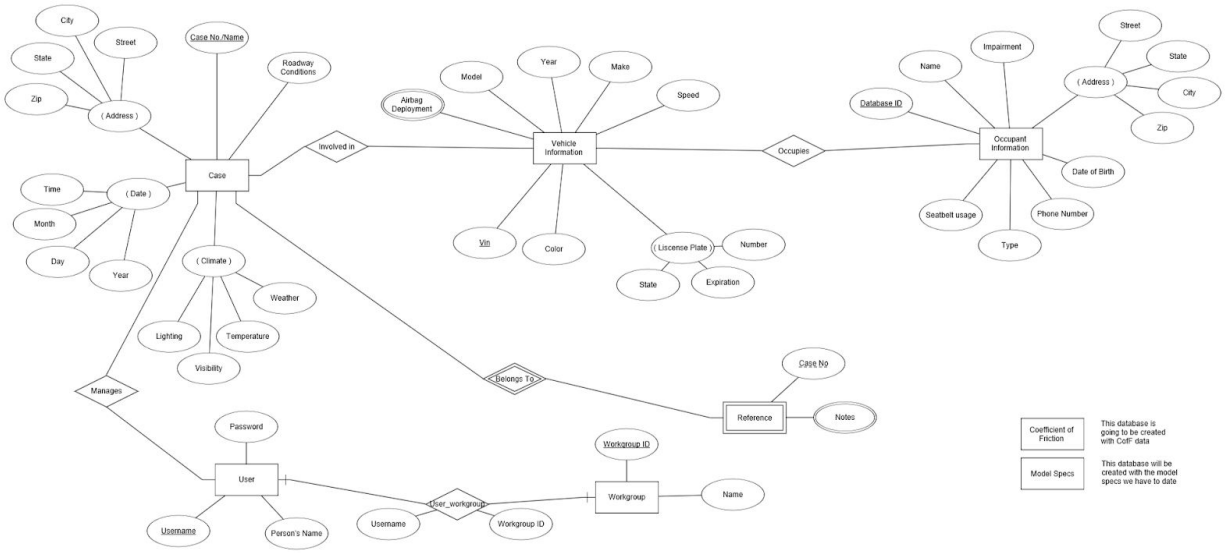
Should something come up or the project prove too time consuming, our team has established a bare bones version that will be completed at the minimum. This version of the database would have the new data included into the database but it will not have the ability for allowing users to add data onto the database. Our enhanced version of this project would include the ability for users to add data and securely store it onto the database.

Conclusion

The requirement specification document is now concluded. This specification will be referenced throughout the project and will be used as a guide when developing the database. More references such as the Entity-Relationship diagram can be found in the appendices below.

Appendix

The following diagram is an Entity-Relationship model created by the database support team last year. It will be referenced when creating new database tables.



Below is a relationship schema that will be referenced throughout the project when modifying the database.

Relational Schema

CASE(Case_No(PK), Address, Date, Climate, Roadway_Conditions, Lead_Investigator, Group)

VEHICLE(VIN(PK), License_No, Make, Model, Year, Speed, Airbag_Deploy, Color)

OCCUPANT_INFO(ID (PK), Name, Seatbelt, Impairment, Address, DOB, Phone_No, Type)

REFERENCE(Case_ID(FK), Witness_Notes, Personal_Notes)

MODEL_SPECS(Make(PK), Model(PK), Start_Year(PK), End_Year(PK), Front_Tire, Rear_Tire, Doors, Body_Style, Drive_Wheels, Curb_Weight, Weight_Front, Weight_Rear, Length, Width, Height, Wheelbase, FOH, Front_Track, Rear_Track, Front_Bumper_Ht, Rear_Bumper_Ht, Ground_To_Base_Windshield, Stability_Ratio, CGH, Stability_Track)

USERS(Username(PK), Password, Pname)

WORKGROUPS(ID (PK), Name)

USER_WORKGROUP(Username(FK), WGID (FK))