

FINAL PROJECT REPORT

Team Members

Divya Sathiyamoorthy (divyas5)

Parul Sharma (psharm42)

Matthew Ghanayem (mg37)

Qiuling Fan (qiuling2)

Changes in Project Directions

Comparing our final application to the original proposal that was drafted in the first stage of the project track, the following changes were made to the project direction:

- Due to time constraints, the user interface of our final application does not reflect some of the details that were mentioned in the project proposal. For instance, we were not able to implement the functionality of the user being able to schedule and/or reschedule appointments
- Due to time constraints, we chose to change the direction of our project from encompassing health, financial, and appointment details of the patient to primarily focusing on displaying health statistics of the patient and viewing information about procedures, diseases, and common conditions
- We were also not able to implement an interactive dashboard with visuals to showcase changes in the health stats of the patient over time due to the complexity of this functionality and time constraints
- The primary reason for the change of direction of our project can be attributed to time constraints and a steep learning curve on how to implement some of the complex functionalities of our application

Application Usefulness

The primary purpose of our application was to simplify the healthcare system for patients and guide them by making important details transparent. With our final application, we were able to accomplish the following achievements and failures.

Achievements

We were able to increase the transparency of healthcare for patients with the use of our stored procedure that allowed the patients to view changes in specific vitals over time. If the patient did not know what was good or bad, our health score outputted by the procedure gave a simple way for patients to evaluate their health.

Failures

Due to time constraints we were not able to implement the visual aspect for our health statistics dashboard, which would have made for a great user interface and technique to display a user's health information.

Changes in Schema and/or Source of the Data

Changes in the Schema

Reflecting upon our current schema and comparing it to the one mentioned in the previous project proposal that was drafted during stage one, we did not have any major changes in our relational schema. The only change made was to the Patient to remove the age attribute that was deemed unnecessary with the existence of the patient's date of birth attribute within the table.

Source of the Data

While drafting the project proposal during stage one of the project, there were ambiguities about the source of the data to populate our database for the project. However, this was resolved since there were two primary sources for our data: the patient information was created using the python library Faker and the data about different diseases and conditions as well as procedures was imported from Kaggle that had some readymade datasets that we were able to import into our database tables.

Changes the ER diagram and/or Table Implementation

When we were thinking about advanced queries in stage 3, one of them required the creation of new tables for practitioner and insurance. One thing we also did in stage 3 was also significantly decrease the size of our string fields since it was overly big and took too much data in the database. Other than that, there were no non-error correcting changes we made to our database design.

Functionalities

Added

Apart from providing the additional functionality of being able to delete their account, which was not mentioned in the original project proposal, no other additional functionalities were added.

Removed

Due to time constraints, our team was unable to implement the following functionalities:

- Users can schedule/reschedule appointments and view upcoming appointments
- Users can view the charges that are due on their account as well as the charges that would be incurred by various procedures
- Users can view medications prescribed by their doctor as well as additional messages that the doctor may have put in during/after the appointment

Integration of Advanced Database Programs with Application

Technical Challenges

- Matthew: The main challenge I faced was the inability to test things with the frontend. What I mean by this is that whenever I tried to launch our application in localhost, I would always get an error while my teammates did not. While I never really tried solving this issue, I should have tried to get help with it. This did end up being a technical challenge for me since everything in the backend I wrote for stages 4 and 5 were just rough drafts that my teammates ended up having to debug.
- Divya: I had a challenge with displaying the axios response as a table for the view stats, find doctor, and find condition pages since I had some issues with storing the response as a dictionary and then displaying it based on key-value.
- Parul: For the most part, the only technical challenge was to make sure that everyone was working with the most recent updated packages for the frontend on their individual systems because sometimes the package-lock.json files had missing modules that were causing react compiling issues. This may have to do with the way these files are stored within .gitignore on GitHub.
- Qiuling:

Comparing the Final Application with the Original Proposal

Tables and Data: although our tables encompassed the entirety of what was mentioned on the proposal plus a few more tables (insurance and practitioner), the populating of those tables ended up being somewhat sparse mainly due to the sheer amount of tables we had and the lack of non-generated data to occupy them since medical information is private.

Functionality: We hit on most of our functionality goals. The ones we did not hit on were due to the lack of features interacting with appointments and billing. We also did not deliver on the ability to view information about certain conditions because of sparse data insertion.

Future Scope for Improvement of the Application

We think we can improve the structure of our database design to include representation of things like various hospital equipment, hospital bed space, and appointment spaces/slots so we can design new applications that inpatient and outpatient providers can use to interact with the database. Eventually, once we expand our database design enough, we would be able to create enough applications to encompass anything that interacts with healthcare within our electronic health record system.

Teamwork

Division of Labor

Stage 1: All of the work was split pretty evenly

Stage 2:

- Matthew created the UML diagram and updated the stage 2 document throughout future stages and also helped in converting the UML diagram to relational schema
- Parul contributed to the discussion for creating the UML diagram and helped in converting the UML diagram to relational schema and got it approved from TA as well as relayed the changes that were suggested back to the team
- Divya: Contributed to the discussion about developing the UML diagram and got it approved from TA as well as relayed the changes that were suggested back to the team
- Qiuling was involved in the discussion for developing the UML diagram

Stage 3:

- Matthew helped with the DDL commands and created the advanced queries
- Parul worked on setting up GCP and MySQL Workbench as well as created tables within the database based on the DDL commands followed by populating the tables to ensure the presence of relevant data available for the advanced queries
- Divya: Found datasets for tables and formatted multiple CSVs to match the structure of the database tables and worked with Parul to upload those CSVs to GCP
- Qiuling worked on indexing and the analysis for it

Stage 4:

- Matthew wrote the backend (initial draft)
- Parul wrote, debugged, and refactored backend api functions as required as well as tested their functionality. Also, responsible for writing the frontend for the login and sign up pages as well as integrating Axios calls within the frontend
- Divya: Wrote the frontend for profile, find doctors, and conditions page, which were initially displayed as a text box

Stage 5:

- Matthew wrote the trigger, stored procedure, and backend to support it (initial draft)
- Divya: Wrote the frontend for health stats page. Converted the conditions and doctors page to a table format
- Parul worked on fixing login and signup routing to link with other pages as well as worked with Divya to integrate Axios calls for the newly built frontend pages in this stage. Also, wrote new backend apis and updated database data as required. Also, debugged procedure and trigger query syntax.

Stage 6:

- Matthew helped with the project reflection report
- Divya: Contributed to the reflection report
- Parul created the demo video as well as contributed to the project reflection report

Management

Tasks were assigned when we all started working on the checkpoint which tended to be towards the end of it which is poor management, but that is what ended up happening.