

Unified Medical Ecosystem

Stage-4.1: Project Report

- **Please list any changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

We did not make any changes to the original proposal. As mentioned in the original proposal, we built a unified platform for three user bases: patients, healthcare providers, and administrators.

- **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**
 - The application provides an easy-to-use interface that enables patients and healthcare providers to quickly schedule, modify, and track appointments with minimal complexity.
 - Healthcare providers can order and schedule lab appointments for patients, significantly improving the patient experience by eliminating administrative burdens and reducing potential communication gaps.
 - Hospital Administrators gain access to a real-time monitoring dashboard that delivers crucial insights about hospital operations, including performance metrics and financial statistics to support strategic planning.
 - The system offers administrators real-time outbreak surveillance capabilities, enabling monitoring of disease trends and could help prepare for potential public health challenges.
- **Discuss if you change the schema or source of the data for your application.**
 - There was no change from the initial prescribed schema or data source except one change in the address field. We used the same schema that we had normalized at the earlier stage. We combined three datasets as the sources. The only schema change was that we added two additional columns in the patient tables. This is due to the addition of new features.
- **Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?**
 - We had not made any changes to the ER diagram or tables in the final stage except for an address field being added. It remains the same from the initial stage. We had clearly planned out the list of features and database requirements at the initial stages. This had given us a fixed plan with which we had arrived at our ER diagram.
- **Discuss what functionalities you added or removed. Why?**
 - Apart from the previously mentioned functionalities, we also added certain functionalities such as addition, deletion, and update of patient details under both the Admin side and the Patient side as per the previous functionality of only the Admin side.

- We also added the functionality of setting the availability of the doctor, which was previously not mentioned in the functionalities. This functionality allows the doctor to set his calendar availability for the next 10 days and makes the whole slot basis for the patient to make bookings.
- **Explain how you think your advanced database programs complement your application.**
 - We think that the complexity of our application starts with handling a large number of appointments, lab reports, and payments. When handling payment-related workflows, it is mandatory to have advanced support, such as transactions and ACID properties, to make sure that no information is lost. It is also crucial to track all the appointment information consistently so that the application can be a reliable single source of truth.
- **Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.**
 - Venkatesh Palanivel - vp35@illinois.edu
 - One of the challenges we encountered was changes in the schema at the later stage of the project after data had been processed and inserted into the table. Even though we planned for the workflows, in order to keep the application performance within acceptable limits, we decided to change the schema, such as processing the location data to its coordinates, so that we could save query time later. I would suggest future teams consider performance impact as well during the start of the schema design along with their workflows so that the processing time is saved.
 - Vishal Ravichandran - vishal8@illinois.edu
 - One of the major challenges faced in backend development is deploying the backend app to the GCP. Though deploying the backend app to the Google app engine is a simple command, 'gcloud app deploy', it was really time-consuming to figure out the configuration that needed to be set in yaml file and connect the cloud SQL to the Google app engine.

Initially while developing locally, we were using a local IP address to connect the MySQL instance, but for the Google app engine to connect we needed to build a socket path and use that to connect the SQL. Secondly, in configuration, it was important to set up a health check endpoint which needed to be given as readiness_check configuration to let the app know when it is ready for incoming requests.
 - Sabhari Parameswaran - sabhari2@illinois.edu
 - One of the major problems was to read the data from the source, which is often in very different schemas. Most of the data we used for this project was big data, which consists of 50K+ rows and multiple columns. Thus, we need to process this data in a way that fits our table and schema

requirements. Most of the time, these transformations and insertions take about 6 to 8 hours. So, one effective process that can be used later is to perform a batch insert. This can be a little more effective than the row-by-row insertions. Also, use multithreading inserts if you can handle the concurrency with consistency.

- Arunesh Kumar - arunesh2@illinois.edu
 - One of the major technical challenges faced during the UI/UX development was maintaining the different states and data in the UI to be used in later stages and pages due to approximately 20+ components. This can be handled better by using the concept of a “redux” storage component in UI, which can make the application faster by relying on storing the data in UI rather than making many calls to the backend.
- **Are there other things that changed when comparing the final application with the original proposal?**
 - In terms of the changes, only one change is there, which is the concept of a separate login and dashboard for “health officials” after the initial decision to create health officials to view the patient’s condition heat map.
 - It was a very redundant login and database flow, which was, in the end, avoided in the final concept by providing the patient condition heat map on the Admin side with various patient and doctor metrics being used on the Admin side to show the advanced queries.
- **Describe future work that you think, other than the interface, that the application can improve on**
 - We could integrate it with the lab reporting software, or we could bring that lab workflow to this application so that it is easier for the hospital management.
 - We could have alerts to doctors’ patients based on the appointments.
 - We could have an advanced analytics platform integrated into this application to analyze the trends and symptoms across the country.
- **Describe the final division of labor and how well you managed teamwork.**
 - Each team member focused on one or two aspects of the project. Venkatesh took care of initial milestones where documentation/report and database design were primary deliverables. Sabari took care of database design/optimizations and managed GCP for our team. Vishal took care of the initial proposal report and the backend of our application. Arunesh took care of the front end and led the team to final completion within the deadline.