IS 436 - Structured System Analysis & Design

**Deliverable 5 - Closing:**

**User Interface Design, Program design and System Implementation**

Team Members:

Adam Afilaka: Developer/Programmer (aafilak1@umbc.edu)

Ashley Braun: Business Analyst/Legal Compliance/Project Manager (410-991-1496; abraun1@umbc.edu)

Zaid Islam: Cyber Security Engineer (240-688-5964) (zislam1@umbc.edu)

Rithika Sayini: Researcher/Programmer (rithika1@umbc.edu)

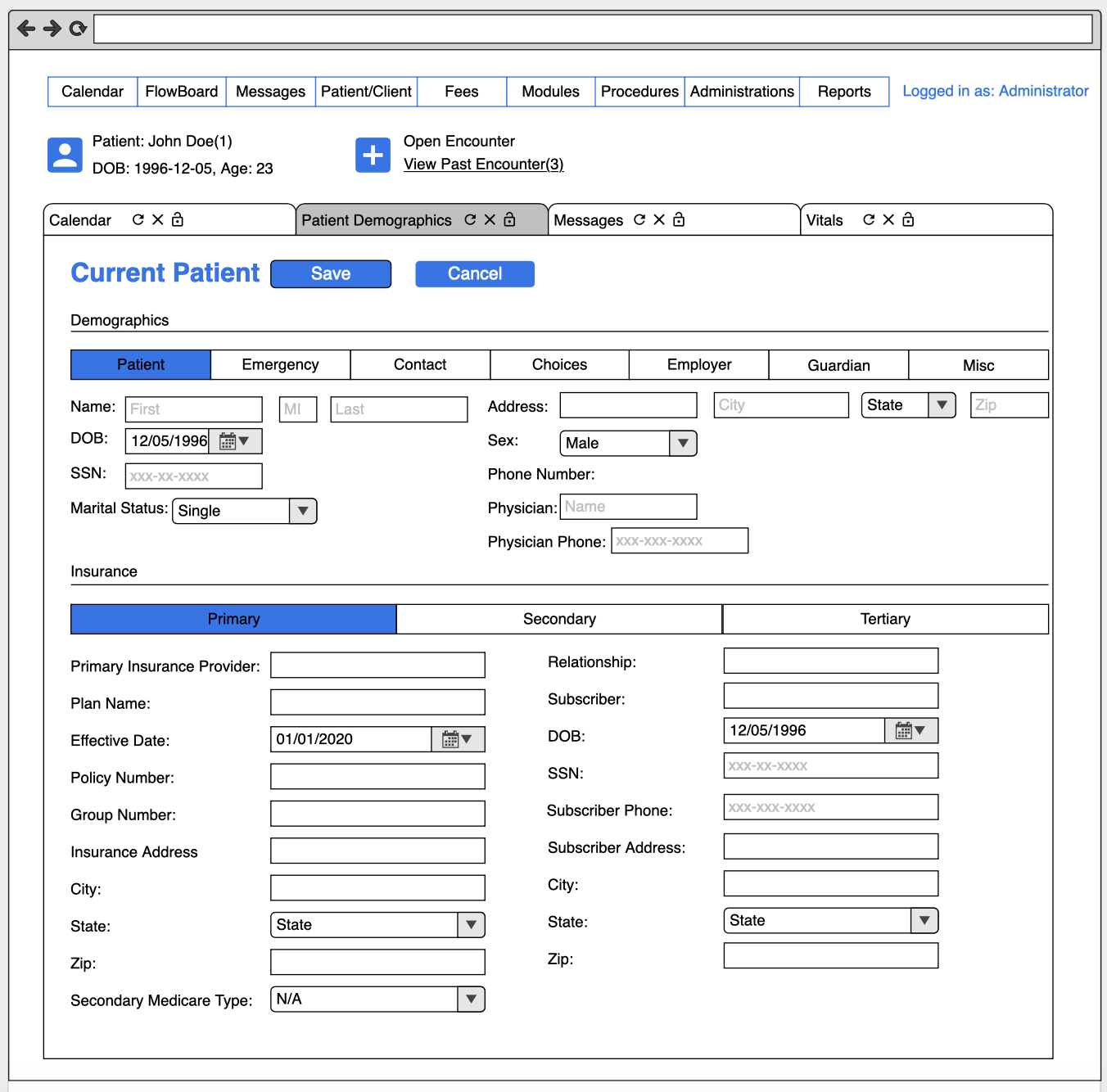
Shanese Scott: Database Administrator (443-760-1459; scsh1@umbc.edu)

Xin Zheng: Quality Assurance(443-931-9403; xinz1@umbc.edu)

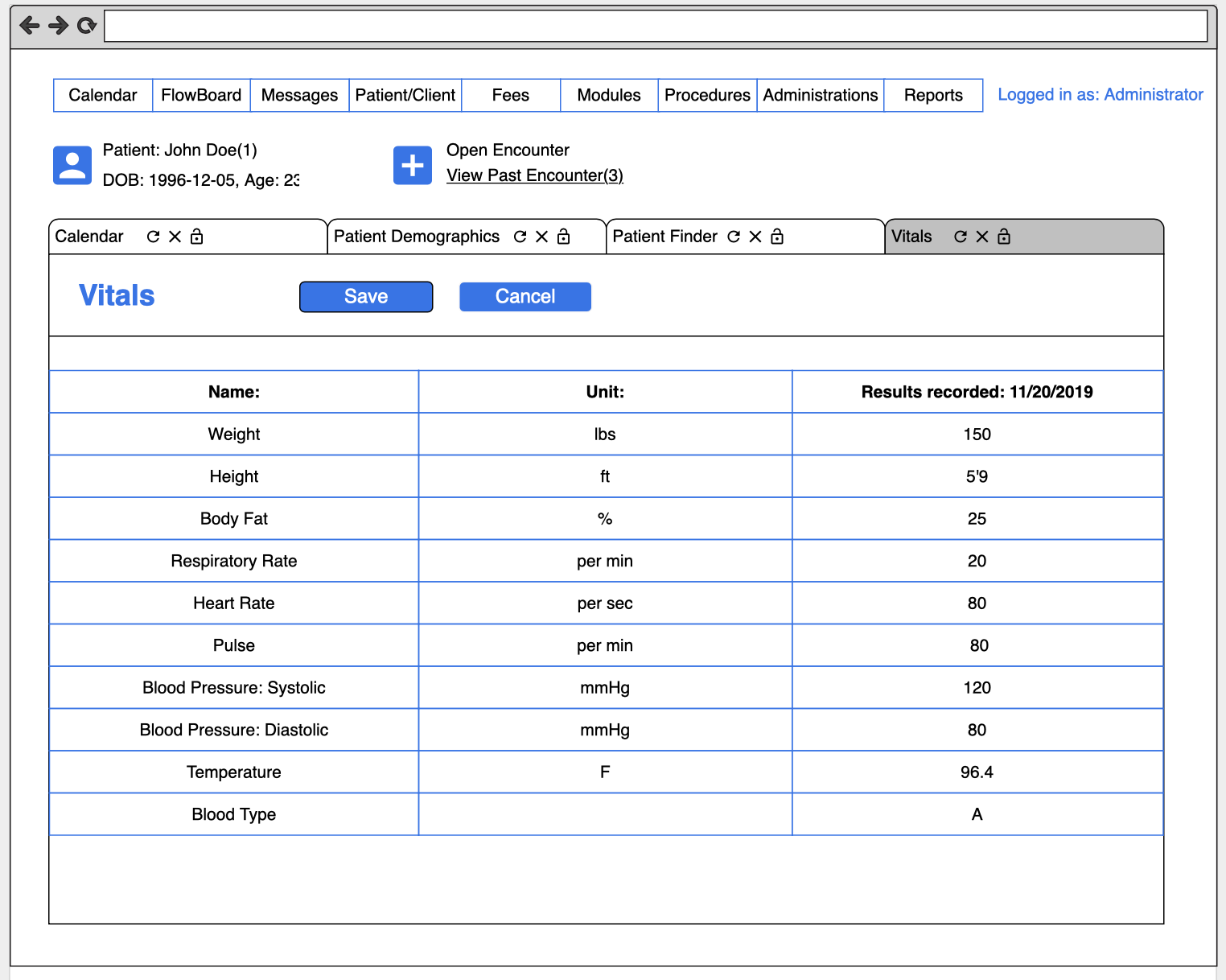
December 5, 2019

**A1 - User Interface Design**

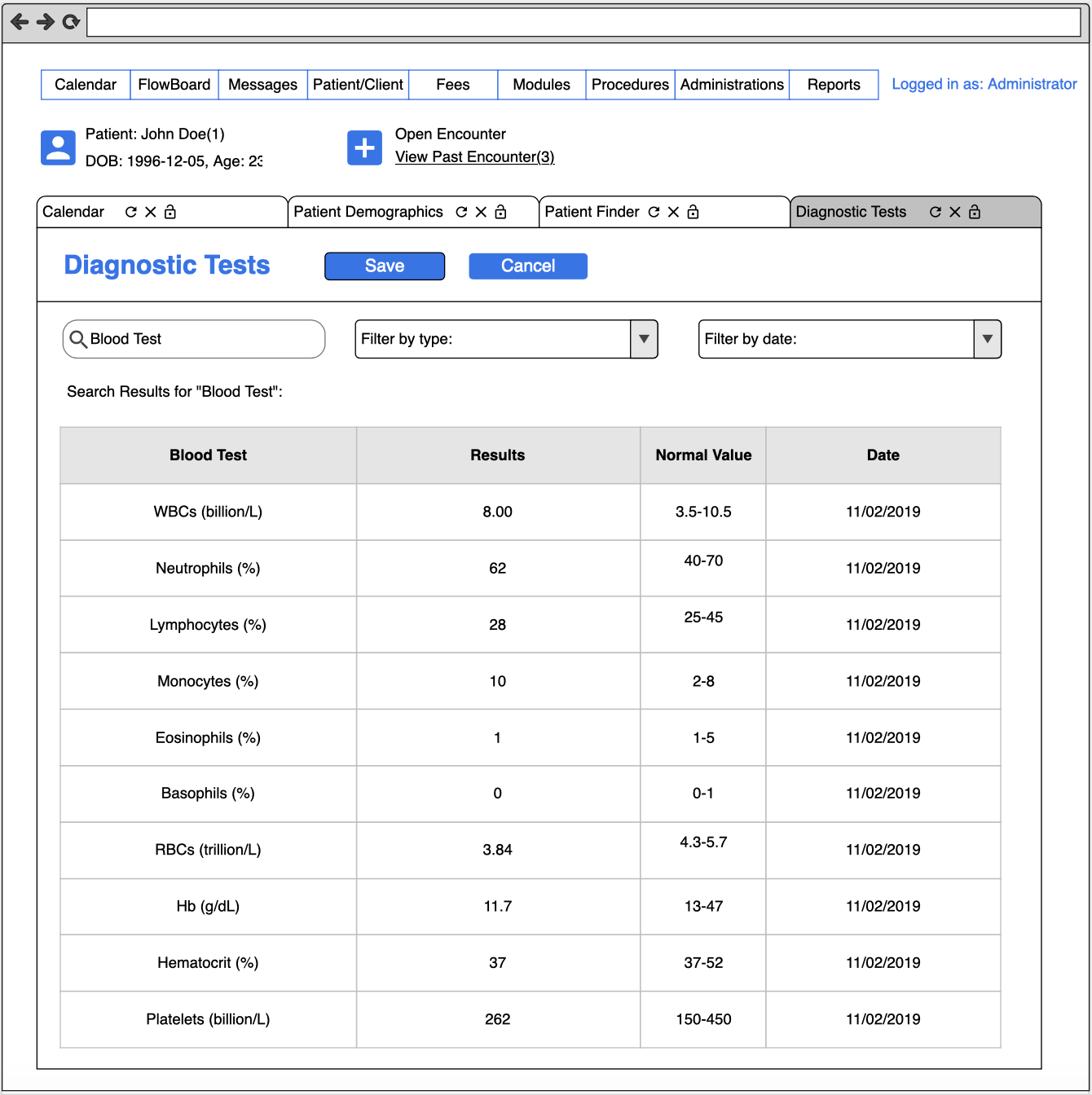
**Patient Demographics:**



**Vitals:**

****

**Diagnostic Tests:**

****

**A2 - Interface Standards**

***Interface Standards to follow and make decisions.***

The capturing of all the below data, in all screens, *Patient Demographics, Vitals, and Diagnostic Studies,* is supportive to healthcare professionals’ critical thinking in emergent decision making, as well as prudent to long-term health.

**Patient Demographics:**

Only a qualified healthcare professional, HCP, be it a doctor, nurse, physician’s assistant, etc. can input patient data into the electronic medical record, EMR. The *Patient Demographics* screen does not trigger events relevant to healthcare treatment, but may require or request a patient’s information be updated from time to time for accurate calculations (e.g., age, weight and height for BMI). This information provides data that is present and needed in the *Vitals* screen to make further analyses and decisions.

Additionally, this interface must display current, up-to-date insurance information, so it requires the authorized HCP to update these fields for proper medical coding and coverage of services.

**Vitals:**

Vitals information is continuously monitored through the wearable device and transmitted into the patient EMR for recording. If data is within normal limits, WNL, data is recorded for a specified period of time and then disposed of once the data is labeled as insignificant. HCPs can also input vitals obtained during a visit to capture a patient’s medical history and for further patient monitoring. In the scenario of manual input from an HCP, data is not disposable. It is recorded and used as a future reference to track the patient’s overall well-being, and capture irregular variations, so when abnormal data is admitted to the EMR it can be analyzed over time to evaluate for actions required, rule-out causes and make further decisions.

When vitals are abnormal and pose imminent danger to the patient’s well-being a signal is sent back to the device to alert the patient. The system may also notify certain physicians of alarming patient data requiring intervention. When data is manually input at an abnormal range it can also alert the physician of certain actions needed to stabilize the patient.

**Diagnostic Studies:**

The *Diagnostic Studies* screen is restricted to input of radiological, pathological, and laboratory results. Upon entry of data, diagnoses and impressions are generated by the system and will trigger a possible follow-up requirement (e.g., “see Cardiologist”, “see Pulmonologist”, “see Endocrinologist”, etc.) or a possible treatment plan (e.g., “take 2 baby aspirin daily”) with a final review of the treatment plan by the HCP.

**B - Program Design:**

Pseudocode:

If (wearable device detected an abnormal){

Warming send to patient and doctor,

Record the vitals of abnormality,}

Else{

Record vitals to database for future use}

If( doctor request medical records){

If(doctor credentials is verified){

Send the medical records,

Else

deny the request}

Update Vitals:

For every 10 second interval the vitals of patients are taken and previous vitals are stored in the cloud in a loop

Store Patient data:

Patient data such as vitals and demographics are sent to the cloud

Call Emergency services

Radio channels as well as cell phone towers are contacted due to the parameters of the vitals regular vitals for said patients are out of range.

Update EMR

https://swe.umbc.edu/~scsh1/ehrsystem/Tab.html

**C 1&2 - System Implementation:**

***Note: C1 is uploading program code to Github. Below outlines C2 requirement.***

**C2-**

1. Software Requirements:

* MySQL
* OS (Linux, Windows, and macOS)

1. Hardware Requirements:

* 16-64 GB of RAM
* JDK 64-bit
* Disk Space 250 GB or higher
* 8-16 core processor
* High speed drive (SAN)

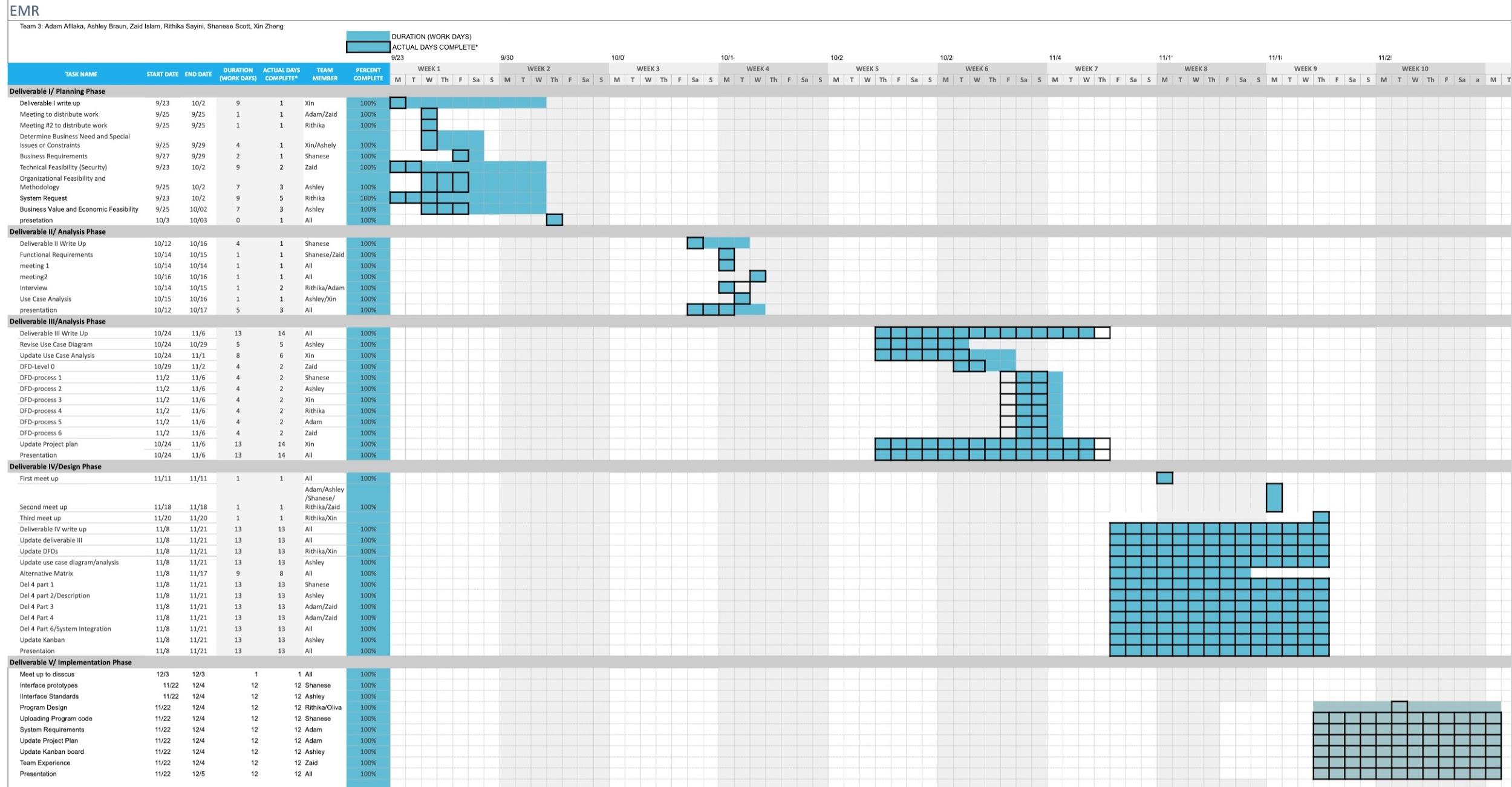
C. Server Requirements:

* 32 GB

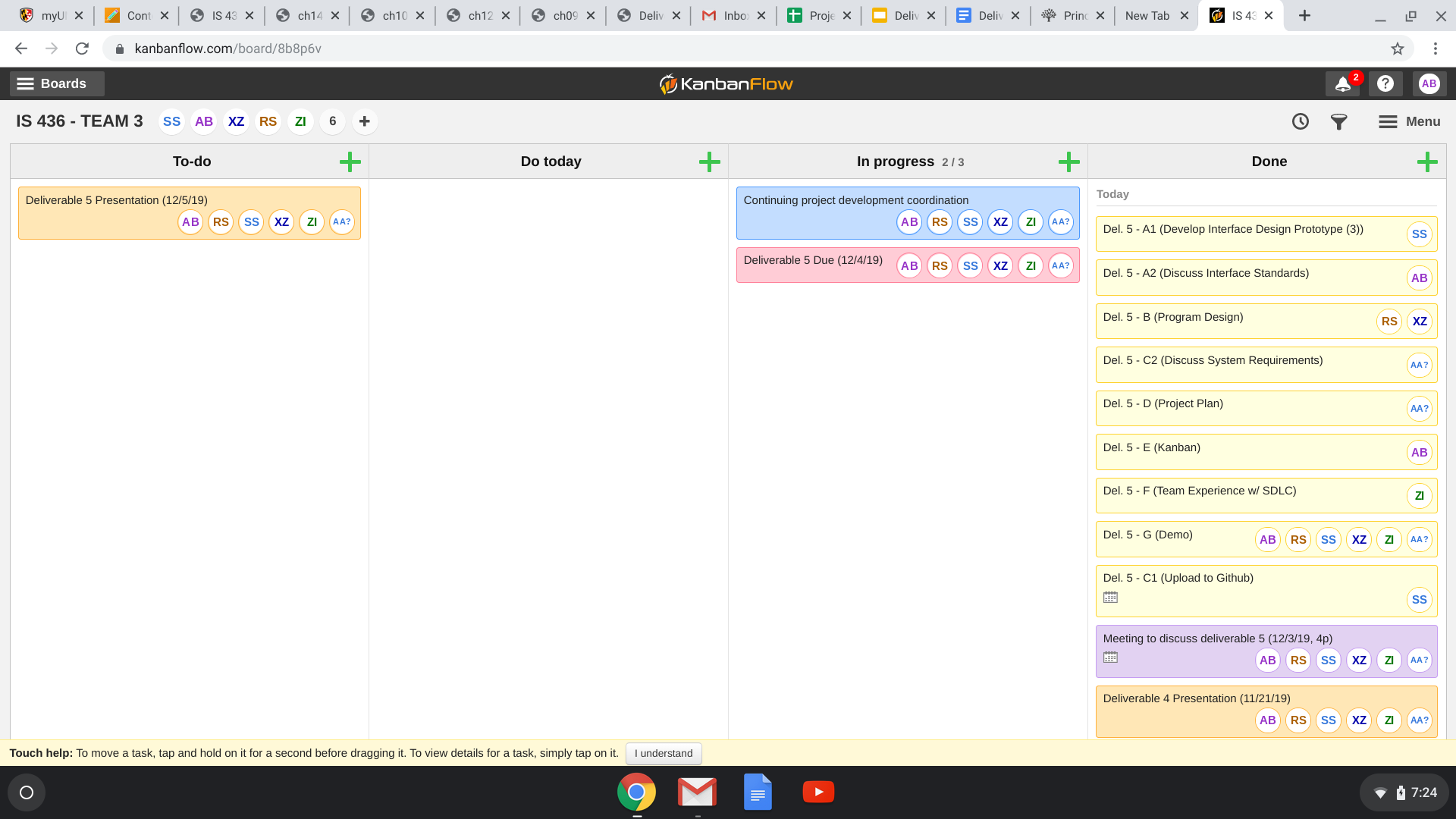
D. Database Requirements:

* Backup Compression
* ACID Properties
* Replication
* Ordered key-value store
* Layers

**D - Project Plan:**

****

**E - Kanban:**

****

**F - Team Experience Through SDLC Process:**

When researching the requirements for this project, a great asset for our group was the background many of us already carried. For instance, Rithika had a lot of knowledge of Health Organization and how a lot of functions work around the procedures in the medical field. We knew that our EMR devices would need a lot of transfer of information and data from two important storage places, and Ashley had a good amount of experience managing databases. With this asset with the group, we knew from the start a lot of the requirements we would need for this System. Researching with organizations helped us understand on what specifications were needed in collecting health information from a patient. Looking at previous EMR concepts allowed to work in innovating to meet our requirements. By looking into our skills and backgrounds, we came together to figure out what we all wanted to produce. Speaking in terms of difficulty, The analysis phase was a little difficult to conceptualize at the beginning. Many of us had different ideas on how this system would be working, so we had to focus on what procedures and materials would, as a unit, fulfill all the requirements we set out. Back and forth from the requirements phase was common. Meetings mostly were discussing how the requirements need to have additional needs. But in the end, we all came to a common understanding on what needed to be done.

During the Design phase we had a bit of difficulty mostly with trying to find out how information would need to be transferred with precision and accuracy from a remote location to a dedicated storage device to be accessed by a Doctor. Designing the Data Flow Diagram as a unit helped look into each specification we needed in our project. After a level-0 was concepted together, each person was able to split up and divide the work pretty well with everyone focusing on each process. Later coming together and seeing how it works and connects together. It was a challenge to fill in the gaps when coming together, but after connecting as a whole, we were able to conceptualize exactly what we were going to produce. In order to create the project, we relied heavily on our DFD charts. It was the biggest help to turn thoughts into practice. We took a lot of time to design the system with its devices. But what made the development phase work so well was the dedication the team put into creating the Design. Difficulty usually only came when needing to see how some details needed to be edited to be done in a more practical sense.

Overall, the team through trial and tribulations found the best and fastest way to solve any issue, big or small, was to communicate fluently and effectively and to not move forward in a step or phase until everyone is on the same level and understanding with each other. We learned the SDLC’s main focus is to work in phases, collaborate with a group, and ensure that as a unit a system is developed with a lot of perspective and diversity and to continue with everyone not just a singular person.

**G - Demo Application**

Demo was presented to our previous interviewees, per below.

Interviewee: JHH Employee

Position: Applications Coordinator for Epic Systems

Interviewee: Senior Physician

Position: Occupational Medicine treating physician in Stockton, CA