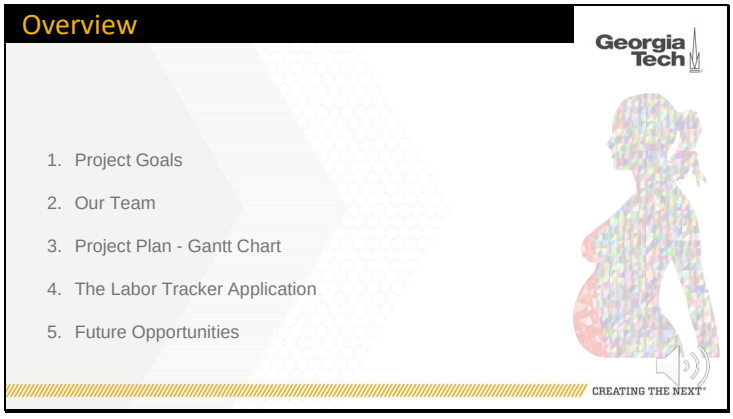
Slide 1

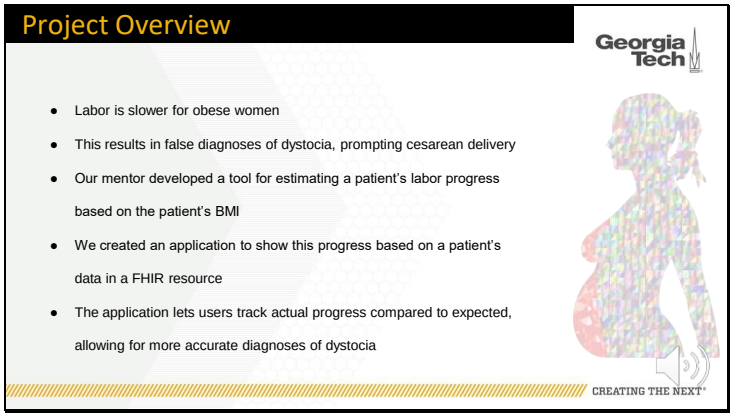
We are team Blue FHIR, and our project for Fall 2017 is The Labor Tracker.

Slide 2



For this presentation, we'll be summarizing our project from this semester. We will discuss the contributions of our team members and our overall project plan. We will then demonstrate the Labor Tracker application itself and discuss some possibilities for the future of the application.

Slide 3



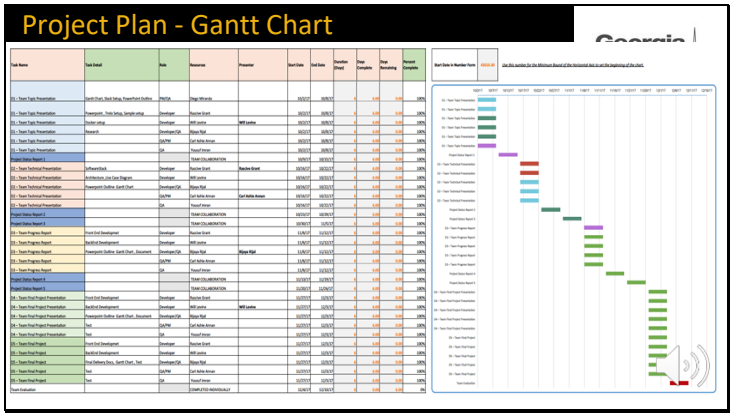
Labor progresses more slowly for women who are obese. As a result, they are more likely to be falsely diagnosed with dystocia, as the providers might believe that she is not as far along as she should be. This can prompt a cesarean delivery, which is much more invasive, dangerous, and costly. Our mentor, Nicole Carlson, developed a tool for estimating a patient's labor progress in terms of cervical dilation over time based on the patient's own BMI, called a partogram.

Our application was built to show this custom partogram based on a patient's data from a FHIR resource. The application lets patients and providers track the actual progress of labor and compare it to the expected progress based on the patient's BMI. By using this custom chart as opposed to a standard partogram, providers may be less likely to resort to cesarean deliveries.

Slide 4

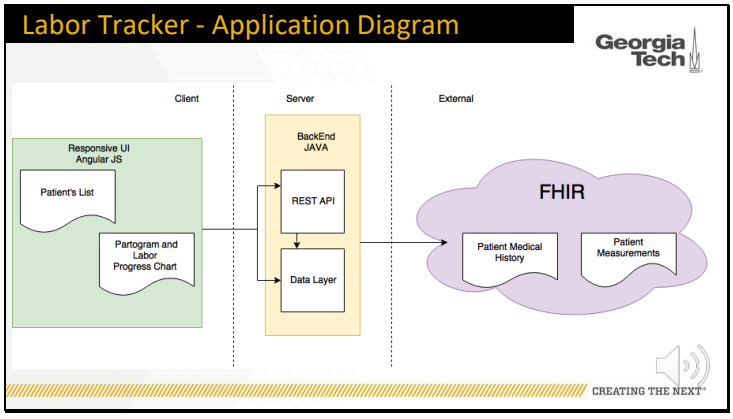
We began the project with a team of six, but over the course of the semester we were reduced to three. Our final team consisted of Ray Grant, who was our PM and primary front end developer, Bijay Rijal, who was our lead technical writer and also performed quality assurance, and Will Levine, who was the primary back end developer and also performed quality assurance.

Slide 5



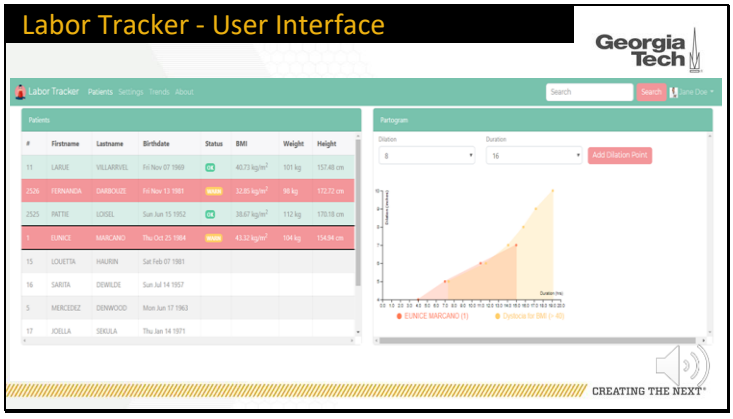
Our team had to adjust and reduce the scope of project after some of the members dropped the class. Wil and Ray worked as primary developers while Bijaya worked on the testing and documentation. After our discussion with the TA mentor and regrouping, the team worked really hard to get the project back on the schedule as described in the Gantt chart. We have been successful to get the minimum target completed in time.

Slide 6



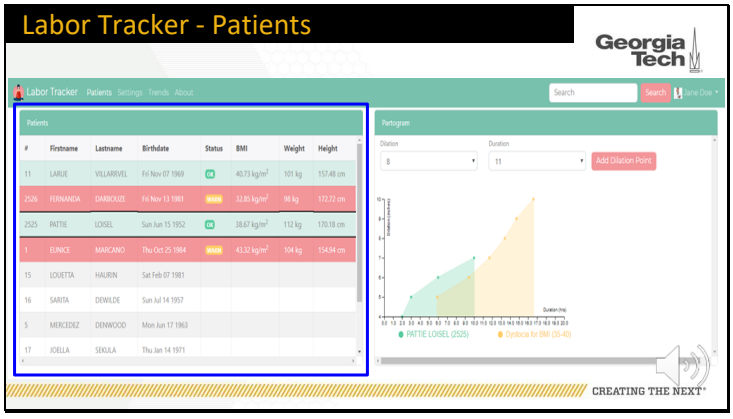
The application consists of three parts, the front end client, the back end server, and an external FHIR resource. Our front end client is built using AngularJS. It communicates with a Java web application server get the data it needs to display and to store input from users. This server in turn pulls patient information from an external FHIR resource. We accessed the EXACT dataset through a FHIR service provided by Georgia Tech, but this could potentially be pointed to other FHIR services using the second DSTU version of FHIR. Our back end server caches the patient information pulled from the EXACT server after it is first requested in order to reduce the number of calls to the server.

Slide 7



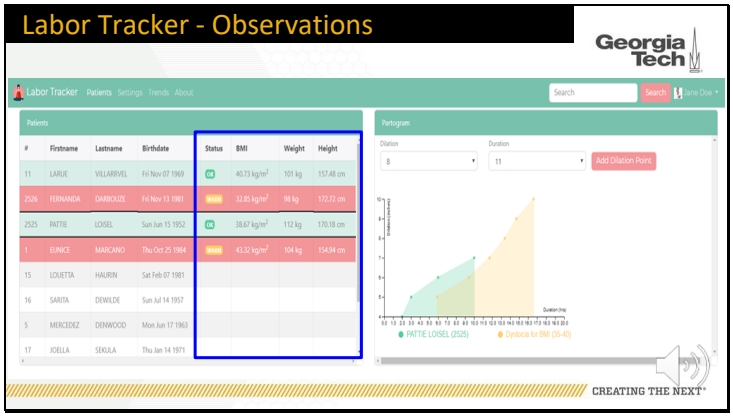
Here we see the main interface of the Labor Tracker application. The options across the top are not functional and purely for demonstration purposes. The two views below it, however, are functional.

Slide 8



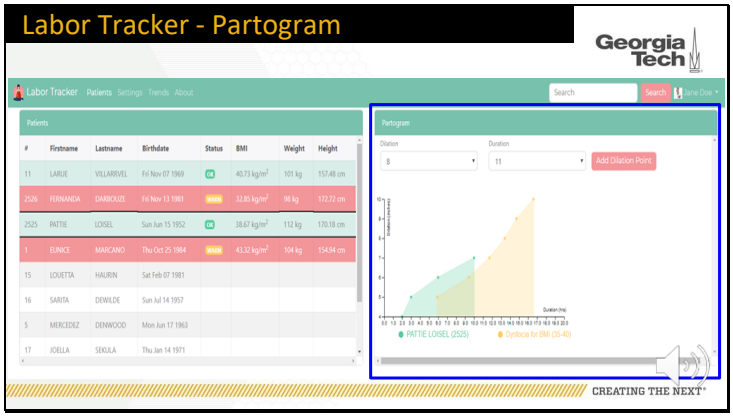
On the left is the patients table. This shows the data for the patients in the application as it was retrieved from the FHIR server. The first four columns show the patient's FHIR resource id, name, and birthday.

Slide 9



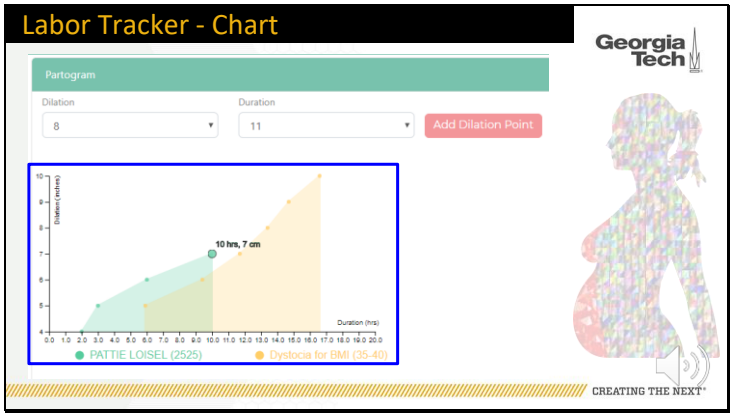
Scrolling to the right in this table shows the last four columns. These show the Height, Weight, and BMI of the patient as well as the most important column, status. The status column indicates whether or not the patient is progressing according to their predicted partogram or if she is falling behind. If she is on track, meaning that her dilations are progressing as quickly or quicker than predicted, the status will show a green “OK” notification. If she is behind, meaning that her dilations are progressing more slowly than predicted, the status will show a yellow “WARN” notification.

Slide 10



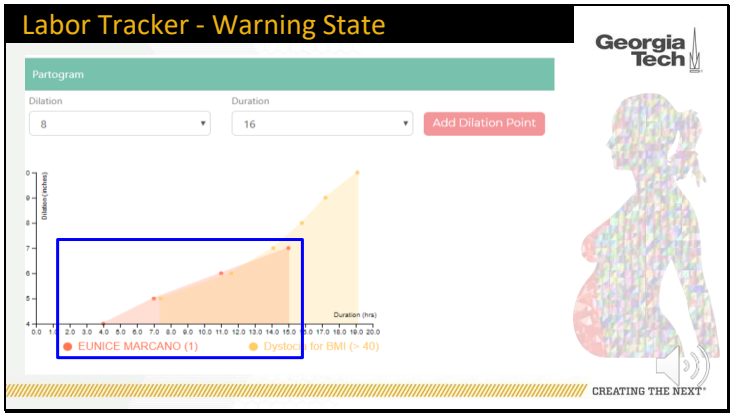
The partogram view on the right shows the selected patient individualized partogram, based on her BMI. This is demonstrated by showing expected dilation over time vs the actual recorded dilations over time for the selected patient.

Slide 11



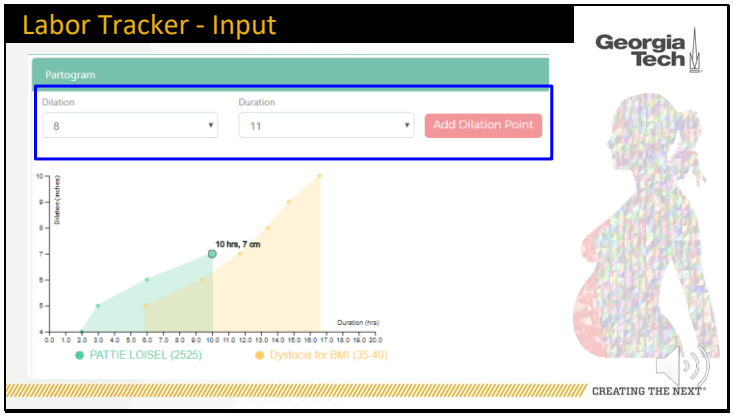
Focusing on the chart itself, we see that the yellow series represents the expected partogram while the green series shows the actual progress of the patient. The duration along the x axis is measured in hours and the cervical dilation along the y axis is measured in cm. Hovering over the datapoints displays the number of hours and centimeters at that point. The diagram here indicates that the patient is progressing well, as her cervical dilation is above the expected dilation for the amount of time she has been in labor.

Slide 12



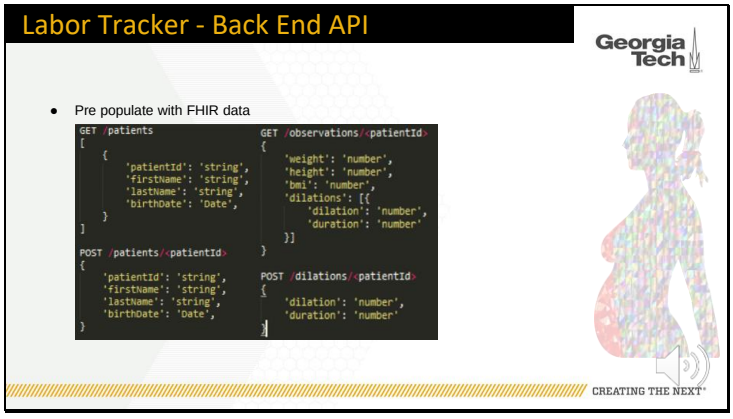
The chart is rendered based on the currently selected patients state. It show a red series for patients in a warning state and green series for patients that are progressing well.

Slide 13



We can track the patient's progress by entering her measurements using these inputs. We simply select how many centimeters she is currently dilated and then the number of hours into labor the measurement was taken and then add the data to the chart using the Add Dilation Point button.

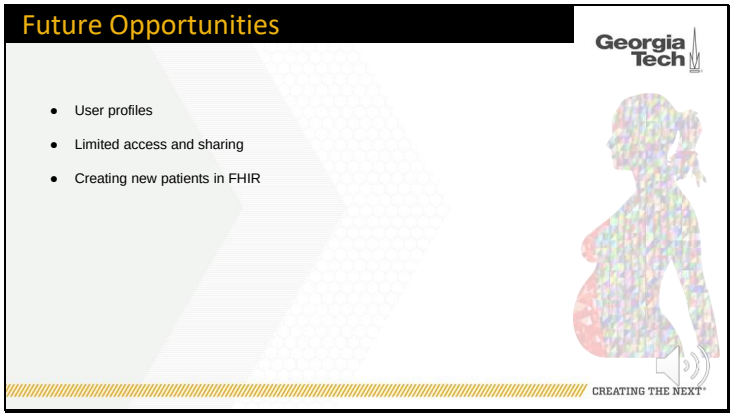
Slide 14



When the back end of our application receives its first request for data, it makes a connection to our EXACT database FHIR service, provided by Georgia Tech. It then queries the FHIR service to get a list of patients using a hardcoded list of IDs. We then extract basic administrative data from the Patient resources returned from FHIR such as names and birthdays. We then query the FHIR service for height and weight observations for each patient. We then cache the patients and their measurements in memory for faster access.

Our back end Java server provides data to our front end via an API. The front end communicates with the backend using GET and POST HTTP requests. To get one or all patient records, excluding their BMI measurements, the front end makes a GET call to /patients or /patients with the specific patient ID as a path parameter. To get their heights and weights for calculating their BMIs, it makes GET calls for each patient to /observations with the patient's ID as a path parameter. To record a dilation measurement, the front end client makes a POST call to /dilations with the patient ID as a path parameter and the dilation measurement and duration time in the POST body.

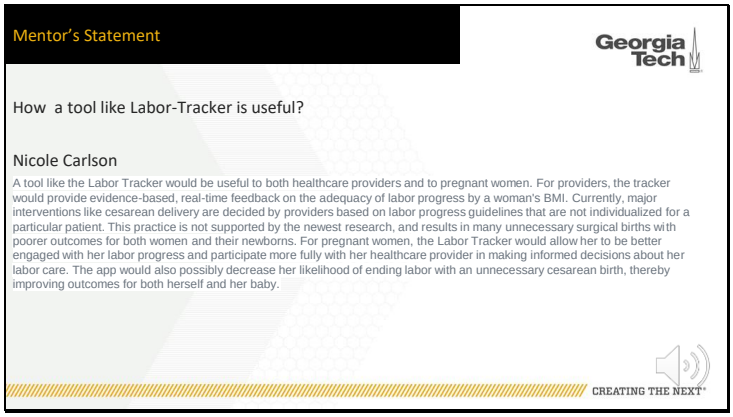
Slide 15



We would like to allow patients and providers to create password protected user profiles to customize their view in the application. In particular, we want to do this to allow users to limit access to their data to only those they wish to share it with. A patient, for example, would likely only see her record and chart in the application and the option to share that record with her provider. Her provider, however, would likely see the records of all of the women the provider is caring for.

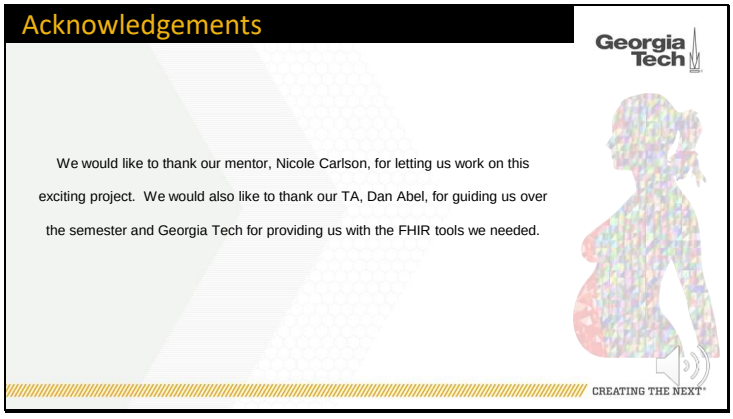
We would also like to enhance our FHIR integration by allowing users to create patients on the FHIR server through our application. This could be both manual and when a user in labor creates her profile.

Slide 16



This is the response by our external mentor as is on being asked the usefulness of the Labor-Tracker application.

Slide 17



We would like to allow patients and providers to create password protected user profiles to customize their view in the application. In particular, we want to do this to allow users to limit access to their data to only those they wish to share it with. A patient, for example, would likely only see her record and chart in the application and the option to share that record with her provider. Her provider, however, would likely see the records of all of the women the provider is caring for.

We would also like to enhance our FHIR integration by allowing users to create patients on the FHIR server through our application. This could be both manual and when a user in labor creates her profile.