Joe Durko (jsd94@drexel.edu), *Kaushik Mukherjee* (km3762@drexel.edu), *Taiwo,Owolabi* (ot22@drexel.edu), *Wendy Prayer* (wp86@drexel.edu)

Assignment 2

Mission

Our mission is to provide better technological solutions for the people working in healthcare by allowing for simpler data retrieval.

Problem

JKTW Medical Center is having trouble storing and retrieving their medical records for patients and their appointments. Patient records are not properly secured in the current Electronic Medical Record system (EMR/EMS). Due to the loose restrictions on patient record entry, the hospital staff consistently has to correct record errors associated with manual client data entry. Furthermore, records are stored in unparsable scanned PDF's organized in directories on a central server. Because of this, searching for records based off patient information requires an unreasonable amount of searching through the scanned PDF files. Simple tasks such as finding the phone number of a patient who needs a follow-up consultation can take sometimes hours and the scheduling is similarly hectic as the schedule is maintained by an employee who updates an Excel spreadsheet weekly. There's also an issue where reporting and auditing takes whole weeks to find the correct scanned forms for certain patients.

Users

The users for this system are the Staff Members at JKTW Medical Center. This includes the doctors, nurses, clerical staff and hospital administrators. These employees require interacting with patient records including their appointments with doctors and nurses. Note that the patient is presumed to *not* have access to this system.

Link to GitHub:

https://github.com/dream-team-se577/jktw_emr/blob/master/README.md

Features

User	Feature Name	Description
Staff Member	Dashboard	As a staff member, I would like to navigate the dashboard Register patient, retrieve patient create appointments and schedule follow-up
Staff Member	Retrieve Staff Information	As a staff member, I would like to retrieve accounts for other staff members currently in the system including contact information, and their role (doctor, nurse or clerical staff).
Staff Member	Update Staff Information	As a staff member, I would like to update staff information in case contact information needs updated or the staff member switches roles.
Staff Member	Register Patient	As a staff member, I would like to register a patient into the system when they visit for the first time including their name, social security number, medical history and contact information.
Staff Member	Medical Record Retrieval	As a staff member, I would like to use a patient's social security number or name to retrieve their information such as lab/diagnostic records, appointment history, and contact information.
Staff Member	Update Contact Information	As a staff member, I would like the ability to update existing contact information on a patient.
Staff Member	Update Medical Information	As a staff member, I would like to update the medical information of a patient including their appointments and lab tests.

Staff Member	Search Appointments and Lab Records	As a staff member, I would like to search for medical records of patients including a date range of appointments and lab records.

These features will remedy the problems plaguing the current system(s) utilized by JKTW staff. Record retrieval can be achieved based off a number of fields such as name and SSN. They can also schedule appointments and lab visits easier by using queries to search within a date range. The search functions will allow for easy recall on patients who have visited previously so that medical staff can view health and contact information.

Use Cases

Name:	Create medical staff in system
Description:	Hospital administrator needs to create a new account for a new doctor, nurse or clerical staff
Primary actors:	Hospital administrator
Secondary actors:	Doctors, nurses, clerical staff
Flow of events:	The use case starts when a new medical joins the hospital. The new medical staff fills in new employee paperwork, which gets routed to the hospital administrator. The hospital administrator utilizes information in new staff paperwork to create an account for new staff in the system
Exceptional flow of events:	N/A

Name:	Edit doctor, nurse or clerical staff in system
Description:	Hospital administrator needs to edit an account for an existing doctor, nurse or clerical staff
Primary actors:	Hospital administrator
Secondary actors:	Doctors, nurses, clerical staff
Flow of events:	The use case starts when an existing doctor, nurse or clerical staff has account properties that need to be updated such as a name change in the event of marriage. The doctor, nurse or clerical staff fills out paperwork to confirm the change which gets sent to the hospital administrator to update the system.
Exceptional flow of events:	N/A

Name:	Retrieve doctor, nurse or clerical staff in system
Description:	Hospital administrator needs to retrieve account information for an existing doctor, nurse or clerical staff
Primary actors:	Hospital administrator
Secondary actors:	Doctors, nurses, clerical staff
Flow of events:	The use case starts when the hospital administrator needs to perform some action on the record for an existing doctor, nurse or clerical staff.
Exceptional flow of events:	N/A

Name:	Utilize Dashboard Search for Patient
Description:	Doctor, nurse, clerical staff have successfully logged into the dashboard and searches for a patient
Primary actors:	Doctors, nurses, clerical staff
Secondary actors:	None
Flow of events:	The use case starts when a doctor, nurse, clerical staff needs to locate a patient within the system and want to search by first name, last name, social security or a combination of all 3. All matches will be returned based on all available matches to search
Exceptional flow of events:	The user is not located within the system at which time the system defaults to the create new patient screen

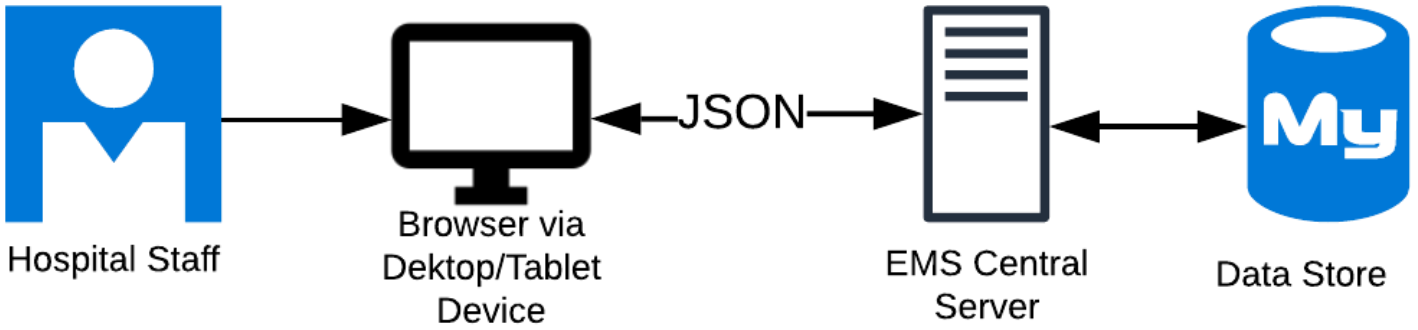
Name:	Utilize Dashboard (Schedule or Retrieve appointment)
Description:	Clerical staff have successfully logged into the dashboard and need to schedule an appointment or retrieve an existing appointment to edit it.
Primary actors:	Clerical staff
Secondary actors:	Patients
Flow of events:	The use case starts when a clerk needs to create a new patient or uses an existing patient's SSN. The clerk uses the dashboard to create or access an existing appointment and then fills in the date, time and the patient's SSN.
Exceptional flow of events:	N/A

Name:	Register Patient
Description:	Clerical staff have successfully logged into the dashboard and proceeds to register new patient
Primary actors:	Clerical staff
Secondary actors:	Patients
Flow of events:	The use case starts when a clerk needs to create a new patient and has the relevant information received from the patient. The clerk will use the dashboard to register the patient's information to be retrieved for later.
Exceptional flow of events:	The required fields are not filled out that constitutes the patient as fully registered. In this case, the patient is not saved into the system and a warning message appears for the clerk.

Name:	Medical Record Retrieval
Description:	Doctor, nurse or Clerical staff have successfully logged into the dashboard and proceeds to search for a medical record by searching for the relative patient
Primary actors:	Doctors, nurses, clerical staff
Secondary actors:	None
Flow of events:	The use case starts when a doctor, nurse, clerical staff needs to locate a patient within the system and know the exact social security number. Using the dashboard, they find the exact medical record of the patient.
Exceptional flow of events:	The user is not located within the system at which time the system defaults to the create new patient screen

Name:	Update Contact Information, Update Medical Information
Description:	Doctor, nurse or clerical staff have successfully logged into the dashboard and proceeds to make update to patient information (Clerical role only) or update medical information (Doctor and nurse role only)
Primary actors:	Doctors, nurses, clerical staff
Secondary actors:	Patients
Flow of events:	The use case starts when a doctor, nurse, clerical staff needs to update a patient record, or a doctor or nurse needs to update a medical record. After retrieving the correct information, they can update a patient's contact information (clerical) or add medical information (doctors and nurses).
Exceptional flow of events:	N/A

High-Level Software Architecture



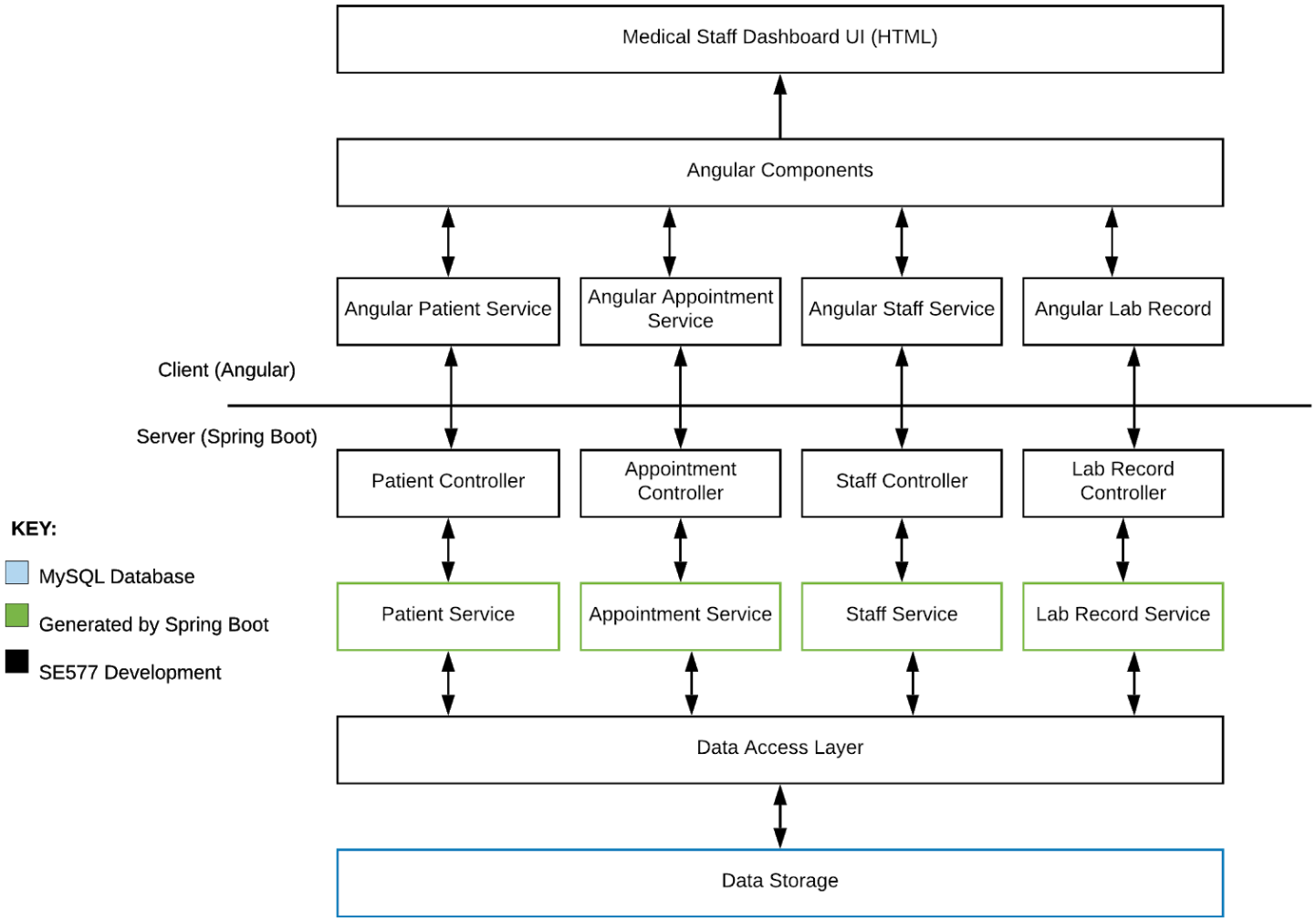
KEY:

SE577 Developed

The JKTW Hospital EMR will be developed in a Client-Server Pattern. Hospital computers and tablets connected to the internal network will use their browsers which will provide the correct forms for creating medical patient records and options to retrieve/search for patient records. Clients will use GET, POST, PUT and DELETE HTTP methods with JSON to communicate with a central Electronic Medical System (EMS) server that is located on a central server machine on the same netwrok. This machine will provide information using a data store on the same machine.

The client library will perform form validation to make sure a record has sufficient information before posting to the server thus preventing invalid entries from being accepted to foolproof record creation. Requests can be filled out via the Dashboard feature which will combine patient searching and record retrieval.

Detailed Layer/Tier Architecture



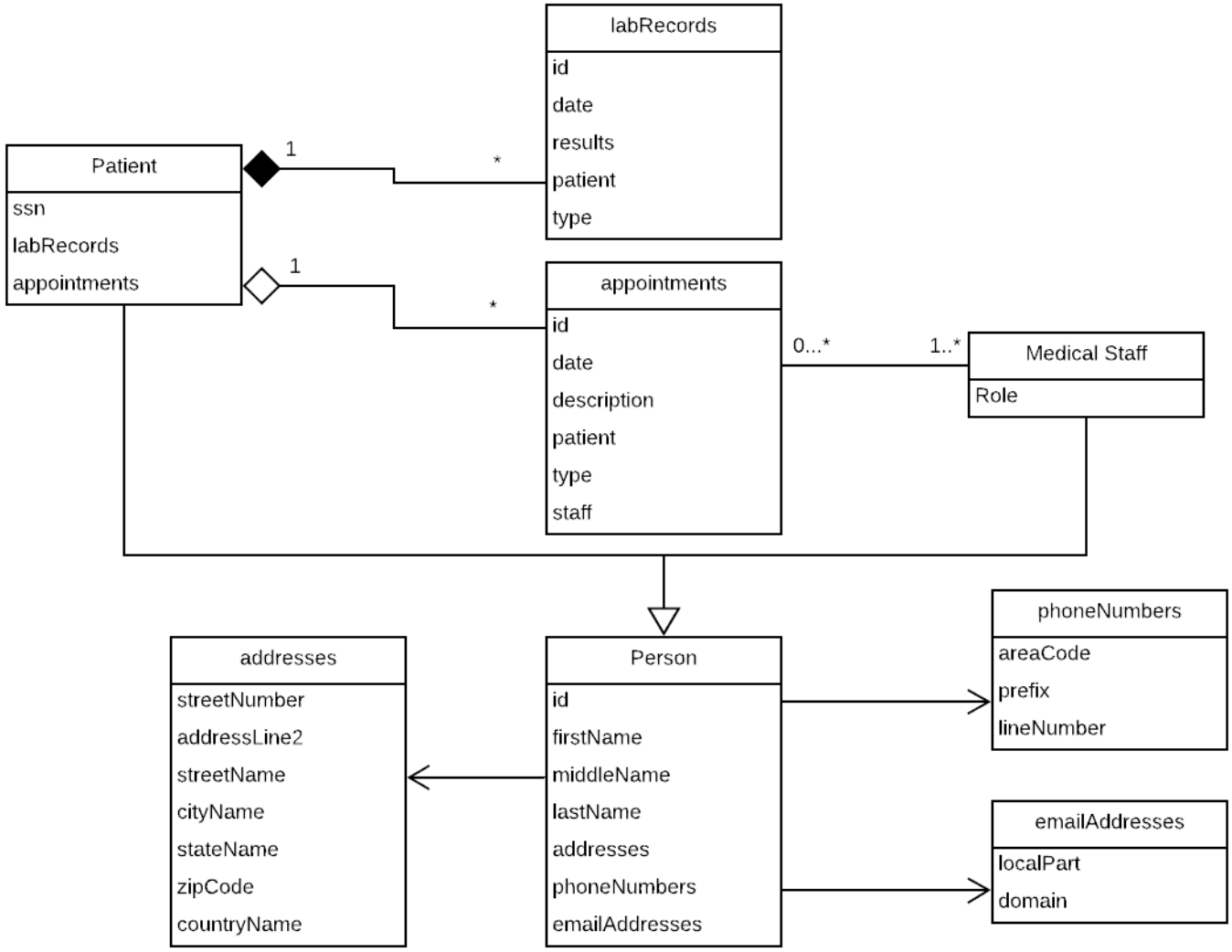
Detailed Layer/Tier Architecture Description

The EMR will consist of a series of layers on the client and server side (tiered between the client machines and a central server machine). A user will interact with the user interface via an Angular-generated webpage in their browser. When the user decides to send the edits of the record to the server, Angular's generated html will make sure all the necessary fields are sufficiently filled out. Next, an HTTP GET/POST/PUT/DELETE calls will be performed against a RESTAPI developed with Spring Boot. The request will then enter a service layer that will validate server that certain properties are not null and converting objects into representational ID's. Finally, a Data Access Layer will interact with a MySQL database that will store the information as a relational database.

The front-end is built in Angular components around the four entity types: Patient, Staff, Lab Record and Appointment. Each of these have five different components (most of which are embeddable in other components):

- **List** [embeddable] - These components will take a list of the entity type and create a table for a quick-look at the entity's attributes and a link to view more details (via a route to the **Form** component).
- **Search** [embeddable] - These components are used for selecting relationships between the entities. Each entity has its own criteria you can search on and it'll query the service to find entries that match the criteria and put them in a table with a button that enables a "Select" which will output the selected entity so the parent component can decide what to do with it.
- **Create** [embeddable] - These components are used for creating a fresh record. Utilizes the **Search** component for resolving references and the **List** component for listing existing picks.
- **Form** - These components provide detailed information on an entity. They can also be used to edit existing entities.
- **Route** - These components are the start pages for the entities that list all existing ones (using **List**) while embedding a **Create** entity at the top.

Data Model


labRecords
id
date
results
patient
type
Patient
ssn
labRecords
appointments
1
*
1
*
appointments
id
date
description
patient
type
staff
0...*
1..*
Medical Staff
Role
phoneNumbers
areaCode
prefix
lineNumber
addresses
streetNumber
addressLine2
streetName
cityName
stateName
zipCode
countryName
Person
id
firstName
middleName
lastName
addresses
phoneNumbers
emailAddresses
emailAddresses
localPart
domain

Data Model

The data model is designed in such a way that allows for medical staff to retrieve relevant information from patient records while also retrieving information of other staff members. There are four different entities:

- Patient (derived from Person)
- Staff Member (derived from Person)
- Lab Records
- Appointments

A patient is an entity that has its own information but also has a one to many mapping to:

- Lab Records are created upon any appointment for an X-Ray, MRI, etc.
- Appointments are created for when a patient checks in for a schedule check-up

Staff members have a many-to-many relationship with appointments since multiple staff members can be on one appointment and multiple appointments can be staffed by the same staff member.

Lab Records exist solely for the patient and do not exist otherwise, and thus have a composition relationship.

Appointments need a patient and at least one staff member.

All of these relationships are mirrored in the database.