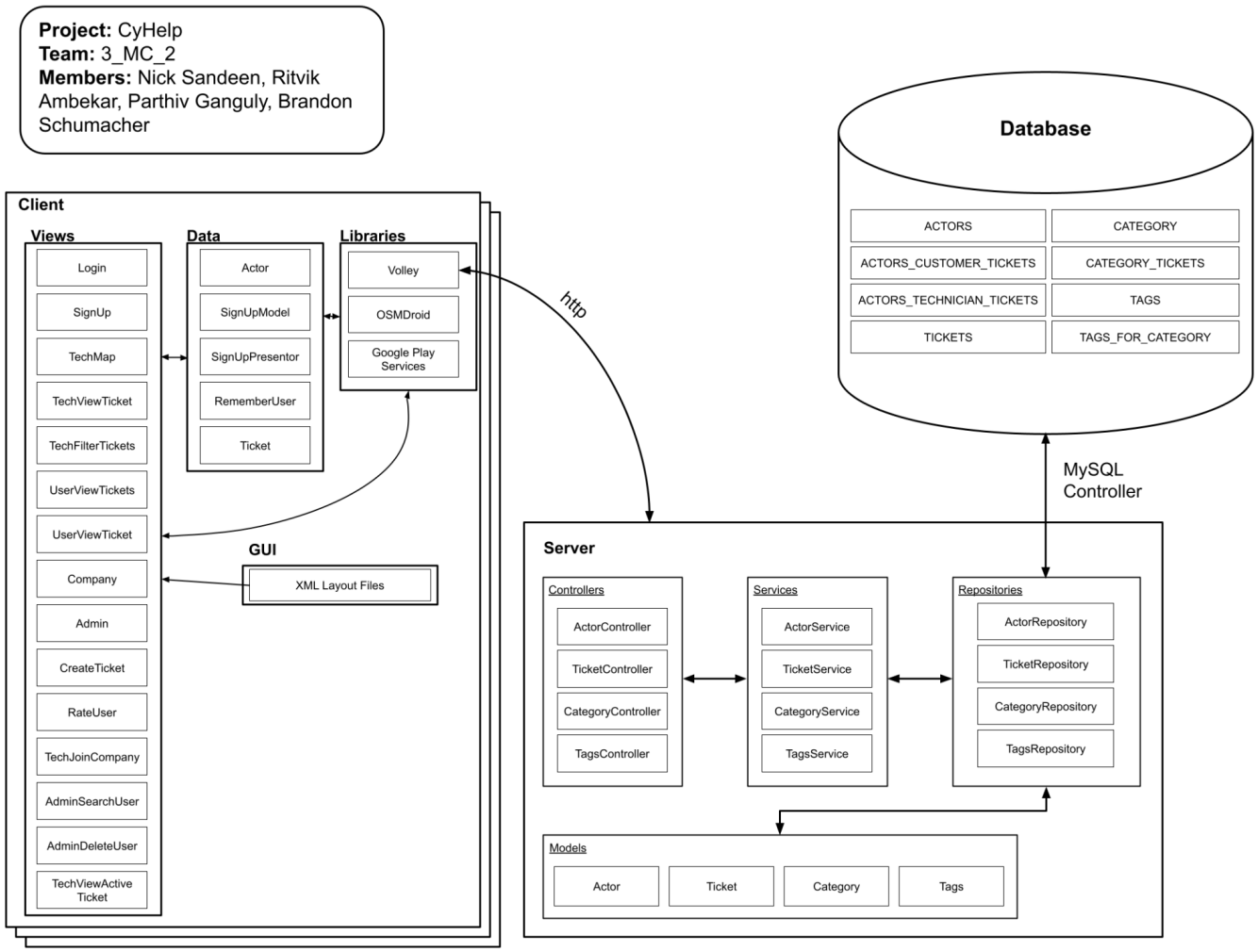
Design Document for CyHelp

Group 3_MC_2

Nick Sandeen: 24.39%
Parthiv Ganguly: 25.61%
Ritvik Ambekar: 24.72%
Brandon Schumacher: 25.28%

Block Diagram



Design Description

Frontend:

Android Views:

The application uses many views that allows the users to interact with the application. The views also handle the interaction between the user input, the server, and what they see on the screen. The views also pull from XML files to show the user relevant information.

Data:

The application uses several data classes which are used for storing objects or performing repeated tasks. For example, the actor class stores the user's info like "user id". This allows objects to be passed between screens and reduces the number of requests made to the server improving performance of the application.

Libraries:

The application design requires the use of libraries such as Volley, OSMDroid, and Google Play Services. Volley is used for sending requests to the server and handling the responses received. OSMDroid is used in the application as a service that shows a map and allows for markers to be placed and have users interact with the map. This is used so technicians can select a ticket which is a marker on the map and view its properties. Google Play Services is used primarily to get the user's location so that it can be used to show the ticket location on the map.

Backend:

SpringBoot Controllers:

The spring boot controllers allow us to connect with the repository and server through the http connection and allows us to interact with the frontend in a seamless way that is uninterrupted and allows us to be quick in processing and gathering information through specialized methods. In the Tags model, you can see this clearly as we allow users to retrieve tags via various ways including by the tag itself or id. This allows users to achieve their tasks without having to sacrifice their line of query and changing it to something that they desire. The controllers allow us to streamline and hand tailor our user's experience while still being effective and less time consuming.

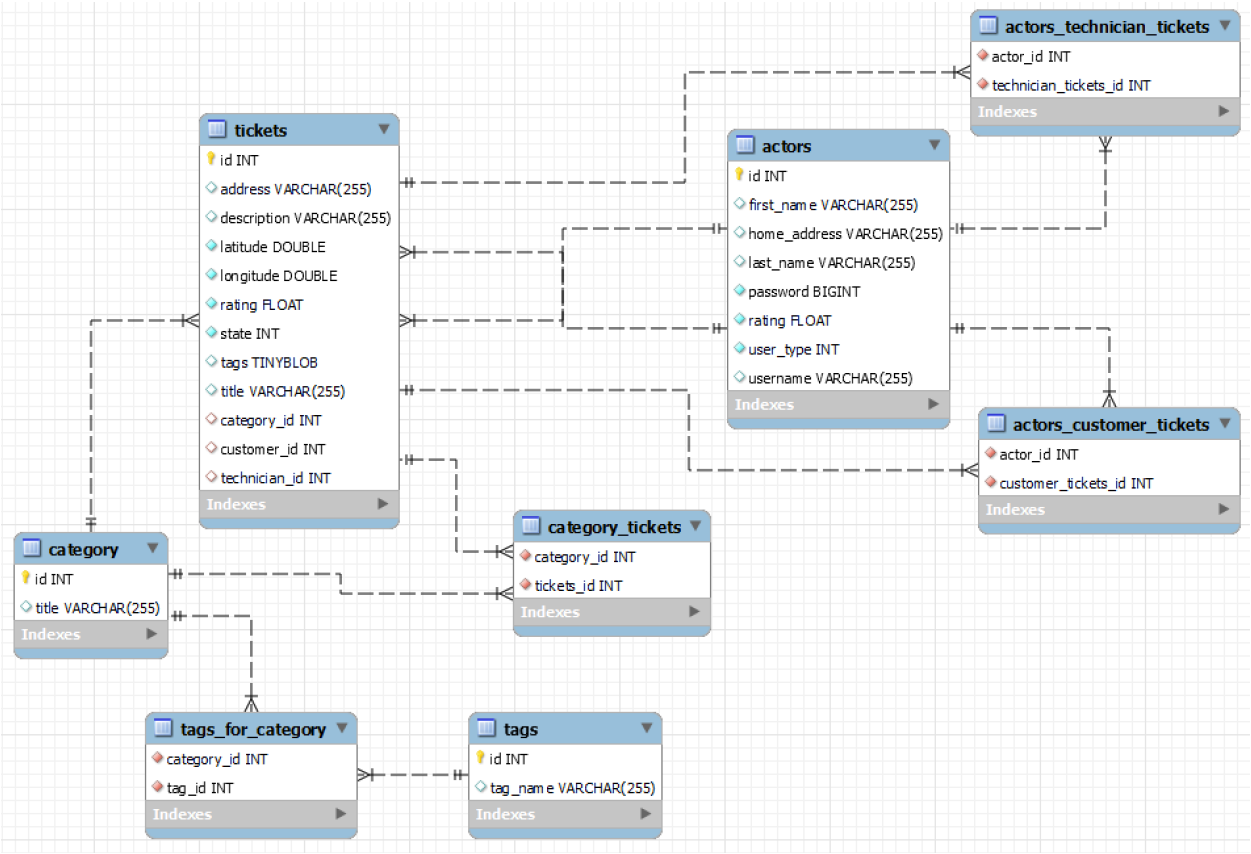
SpringBoot Model:

The spring boot models allow us to implement the key features of the elements that are present in our database. Our Tags and Category models allow us to create diverse elements for our application. The Tags model allows use to create specified tags that our user can create on the frontend and allow other users to easily see the difference in tags. The tags communicate with the Category model and is listed below the Categories to create a database of its own consisting of both the Categories and Tags databases. This connection allows us to create a seamless experience for the user by allowing them to easily navigate, through these methods within our model and the relationship between these two databases, our application and help them achieve their task in an efficient manner.

Spring Repository:

The spring boot repositories are the databases in which we store the JSON elements created from the user and frontend and allow us to store and retrieve elements at any given point.

DB SCHEMA


actors_technician_tickets
actor_id INT
technician_tickets_id INT
Indexes
tickets
id INT
address VARCHAR(255)
description VARCHAR(255)
latitude DOUBLE
longitude DOUBLE
rating FLOAT
state INT
tags TINYBLOB
title VARCHAR(255)
category_id INT
customer_id INT
technician_id INT
Indexes
actors
id INT
first_name VARCHAR(255)
home_address VARCHAR(255)
last_name VARCHAR(255)
password BIGINT
rating FLOAT
user_type INT
username VARCHAR(255)
Indexes
actors_customer_tickets
actor_id INT
customer_tickets_id INT
Indexes
category
id INT
title VARCHAR(255)
Indexes
category_tickets
category_id INT
tickets_id INT
Indexes
tags_for_category
category_id INT
tag_id INT
Indexes
tags
id INT
tag_name VARCHAR(255)
Indexes