Design Document for ISU delivery Service

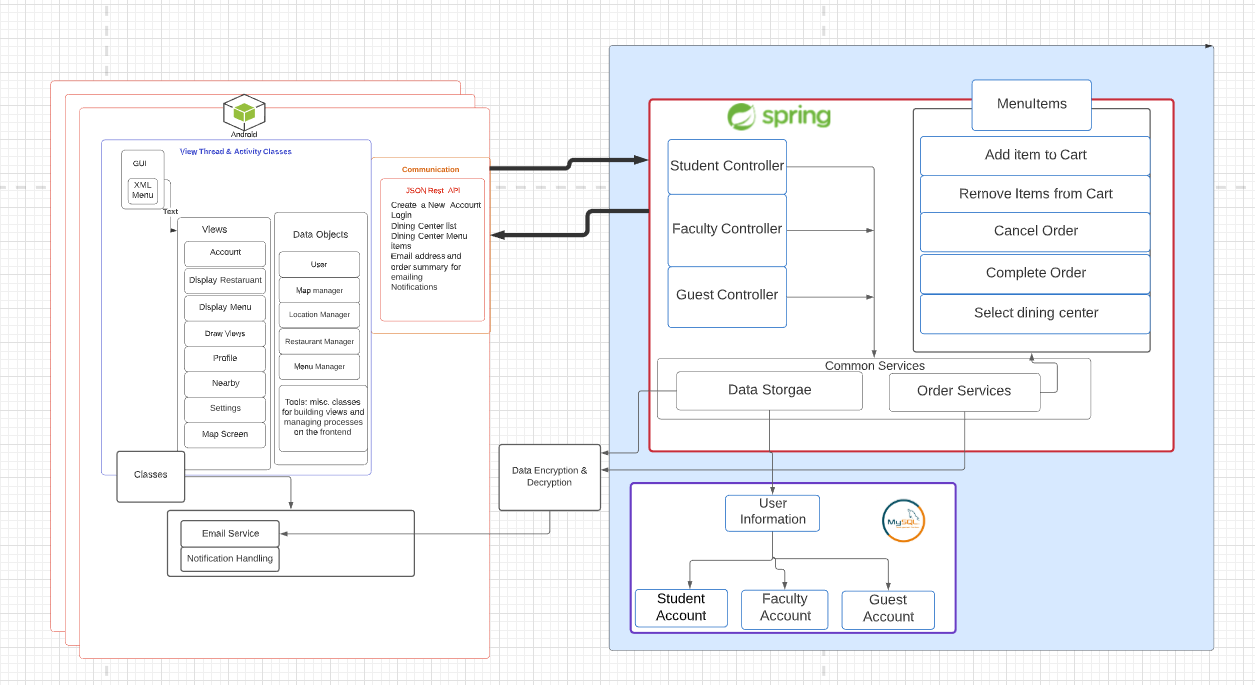
Group SB\_10

Chimzim Ogbondah: 25% contribution

Gerald Edeh: 25% contribution

Omar Almehairbi: 25% contribution

Omar Alsaedi: 25% contribution



**Model:**

Our app utilizes the Google Cloud Platform for both its server-side processing (with

Google App Engine) as well as its data storage (with the Cloud NoSQL schema-less Datastore and Cloud Storage). We chose Google Cloud Platform because of its relative ease of integration with Android as well as its scalability for future use. We use Cloud Endpoints as a RESTful service to ease the communication between our Android client and the API server. We are building off of the Mobile Backend Starter sample code for an Android client communication with a web application backend. This enables us to send push notifications (through Google Cloud Messaging) to notify users about new drawings posted close to them, as well as provide user authentication through existing Google accounts. For the communication with the backend, there will be separate thread which will run in background and will make use of Asynchronous Tasks that will not interrupt the main thread; thus, the UI will be always available to the user even when the background thread will be communicating over the network.

**View:**

Our application’s front-end consists solely of Android clients; thus, we utilize a

combination of XML and Java classes to control the communication and population of our application’s graphical user interface. With DrawNear being data driven, the application needs code written on the front-end to retrieve nearby images and populate views dynamically based on this information returned from the server. The Map view utilizes Google Maps API, which will allow users to view their current location as well as display markers indicating nearby drawings. Other views provide users with the ability to create and view posts, view their profile, and edit settings.

**Controller:**

There are three different controllers which handle the interactions between the frontend and backend. All the controllers Extended the Menu Items class which allows them to use various ordering services. The controllers taking in various request pertaining to storage of user data, order services (add item, cancel order, remove item.. etc) Emailing (utilizes both stored user data and order service data), various notifications

1. Guest Controller
   1. Used for guest users, does not ask for as much user information as the other two accounts
2. Faculty Controller
   1. Used for staff members. Differs for the other two accounts by asking the user for their office address. Gives them the option to have their office as a preset delivery location since it is on campus.
3. Student Controller
   1. Used for current Iowa State students. Is personalized for students by asking for their university ID number. This gives them the option of placing the bill on the U-bill

Shows the relationship between Dining Centers and Items that can be order in the dining center

