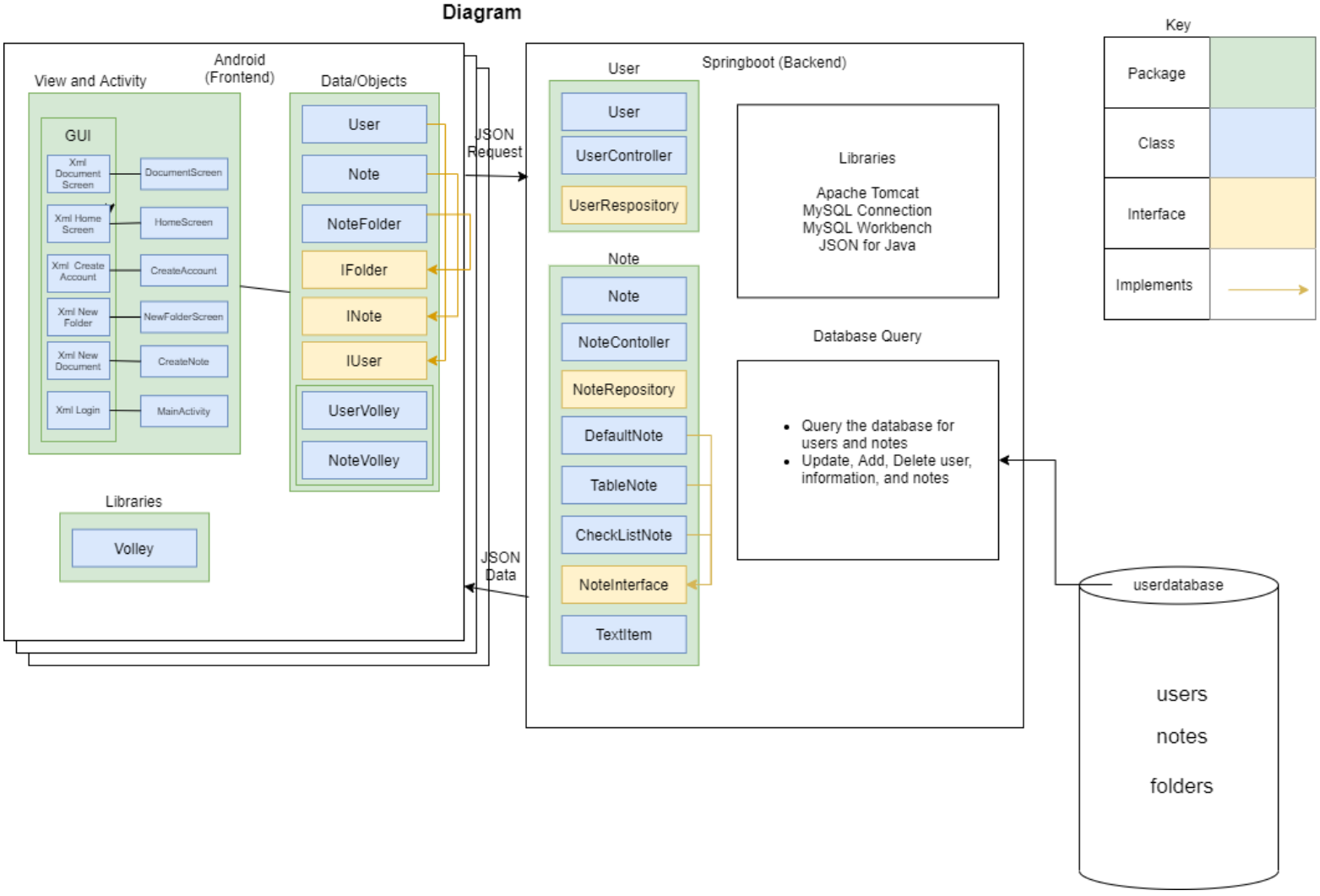
Design Document for Notes With Friends

Group 3_RK_7
Brian Reiss: 25% contribution
Cameron Lettieri: 25% contribution
Nick Pehl: 25% contribution
Coleman Dimmer: 25% contribution

Notes With Friends Block Diagram



In terms of client side, most aspects of design are pretty self explanatory. The activities such as DocumentScreen and HomeScreen all use their respective Xml views and control what happens when those views are interacted with. This can include clicking buttons, or filling out text boxes, they also store what actions should be taken when these events happen. The Data Objects store all the data of each object included in our app thus far. The User, Note and NoteFolder are all used by the activity classes and the UserVolley and NoteVolley are complete with any method needed to receive note or user objects from the server. This helps make the code more modular and adds functionality. The UserVolley and NoteVolley also implement the Volley library which is shown in the diagram under Libraries. IFolder, INote and IUser are all interfaces used to invert dependencies and implemented by NoteFolder, Note and User respectively.

On the backend, there are 2 main things that are stored: user data, and note data. Each user has a name, password, email, id number, and a list of Notes. Note data is a bit more complicated. The Note class takes in a NoteInterface in the constructor. DefaultNote, TableNote, and ChecklistNote all implement NoteInterface, so Note takes in one of those 3 classes when constructing the Note object. Those NoteInterface classes have data and ways to manipulate that data within the note. This is done with the help of TextItem, which can hold a string and a boolean to represent if the item is checked or unchecked. Checked and unchecked status is just not used in DefaultNote and TableNote. The Note class holds a User attached to the note, an Id number, and NoteInterface class. Other than the getters and setters for those 3 fields, each method of Note delegates that work to the NoteInterface class. The User class and the Note class both have controllers and repositories associated with it. The controllers handle all the request and stuff from the frontend, and as the repository to do the corresponding task necessary.

Database Tables