Deliverable #4

STUDENT CONNECT

Kojo Otchere-Badu

Boris Ndoutoume

Christopher Smith

CPRE 339 FALL 2016

Introduction

StudentConnect is a social mobile application that allows students to connect based on their interests. The application features the following: Groups, Events, Books and Notifications. The group feature allows users to create a group and add members. Events allow users to create an event and invite any user. The book feature allows any user to post a book and create a chat to discuss the price. The mobile application requires an account to be created in order to use it and navigate through. Our app is quite user friendly and has smooth transitions between features.

– Two locations where you implement defensive programming

1. We ensure that a user did not have to login into their account again when they re-opened the app, guaranteed that they did not logout from the previous session.
2. We prompt users to enter the right username and password incase they omitted vital information when logging in.

– Two locations where you might have corrected code smells

1. We Created 6 different model objects to represent our 6 main entities. This helps to make a code readable and logical, and avoids us a very large class.
2. We made local variables private to prevent them from being accessed without permission.

– Locations that are responsible for inter-processes/threads communication

Our client application communicates with our server by forming asynchronous HTTP requests. To accomplish this, we created the asyncCall() function in the ServerRequestUtility class. This helps us make GET requests to different PHP files on our server. In turn, our PHP server connects to a database which contains all of our user data.

– Three locations where you implement three design patterns

We use Object Oriented design for the modelling of entities in our application. We thus represent accounts, books, comments, events, groups and notifications.

We use the Client/Server design pattern to allow for users to log in and use the app from different places. Our android application connects to a PHP server and MySQL database.

We also use the MVC pattern through our android application. The views correspond to the layouts, the controllers are our java code, and our models are contained within the tables of our database.