Application 3 Proposal

Assignment Documentation Application

Ryan McManus

David Hunter

10/30/13

Proposal for Application 3

**Project Description**

For application 3, we’re proposing an application that allows a user to track assignments for classes, separated by semesters, all defined by the user. The app will keep track of current and past assignments, as well as prompt the user for grades for each assignment so the application can track the current semester GPA.

**Application Flow**

Our application will begin at a login screen that will allow users to either sign up or log in to the application. The login credentials will be saved on an online source called Parse.com, explained further in the “Saving Information” section of this proposal. The site’s API allows login validation to both create new subscriptions and allow the access of user content by providing a username and password. Also on this screen is a link that allows a user to skip logging in, which leads to several events. Information is saved locally - also explained below in “Saving Information” - for further access on this device, but does not allow multi-device access. If this link is selected, a fragment will display making sure the user knows that information will only be saved locally. This contains a button allowing the user to move forward.

Once a user either creates a log in, logs in, or skips login and accepts the fragment, a semesters activity starts. This is somewhat ambiguous, as it will allow users to choose quarters, trimesters, or semesters. They will display vertically in a ListView, allowing the user to scroll through and select any division. These contain the name of the division, specified by the user, the number of credits (determined by the classes added, explained next), and the GPA. The GPA is calculated based upon grades of assignments, and if any weights are included when the division is created. For example, if assignments are worth 20%, quizzes are worth 15%, participation is worth 10%, projects are worth 35%, and exams are worth 20%, then the application would calculate based upon those weights. Otherwise, everything is taken as the same weight, and divided proportionally.

Once a division (semester, quarter, etc.) is selected, an assignments activity is displayed, and divided based upon the device being displayed on. If the application is on a table, two fragments are displayed. One contains all the classes within the division, and the second contains any assignments associated with that class. If the “All Assignments” view is selected, all assignments are displayed. These assignments display the name, the grade assigned, the priority, date due, and whether it’s pending or completed. If the device is smaller, then two separated fragments are displayed. The first contains each of the classes, as well as “All Assignments”, and the second displays any associated assignments. The views will look similar to the table, but will be separated due to the smaller screen.

**Passing Information**

The information passed from the login activity is whether or not the user has logged in. If they’ve logged in, the Parse API allows us to maintain that login information across all activities. Otherwise, we need to pass that the user skipped login so that information is saved correctly.

The current division selected (semester, quarter, etc.) needs to be passed to the assignments activity so that we can display the division, and any classes associated. Not much information is passed in this application because most of it is saved online or in a local database.

**Saving Information**

We are going to be using the Parse API to save information. This has several advantages: it allows us to save the information in a new way, but also allows a user to save assignment information on one device, and access it on another.

Saving information needs to happen in one of two ways. If the user has created a login, or signed in, that information needs to be validated. The Parse API allows login validation, and returns whether or not the login is valid. This then allows us to save the information online, and return it in any way we need. Because Parse is a document-oriented database, it’s less rigid than a MySQL database native to Android.

Even if the user decides not to log in, the information should be saved on Parse under an assumed user id. The way Parse works is that each device is automatically generated a user id if no login information is generated. If the user then decides to create a login, that information is credited instead of having to be recreated.

The one problem we may face is offline usage. Saving information in a MySQL database is rigid, and would have to be saved similarly to the Parse information so that when Internet access is reestablished, the information can again be saved to Parse.

**Submission Goals**

Our goal is to have all UI information displayed, as well as information input working by the intermediate submission. Integrating the Parse API is easiest when starting a new project, so we may try to get login validation set up by intermediate as well. If that isn’t finished by the intermediate, it’ll be close to finished. By the final submission, all information input will be stored online, or locally if the device is not connected to the Internet.