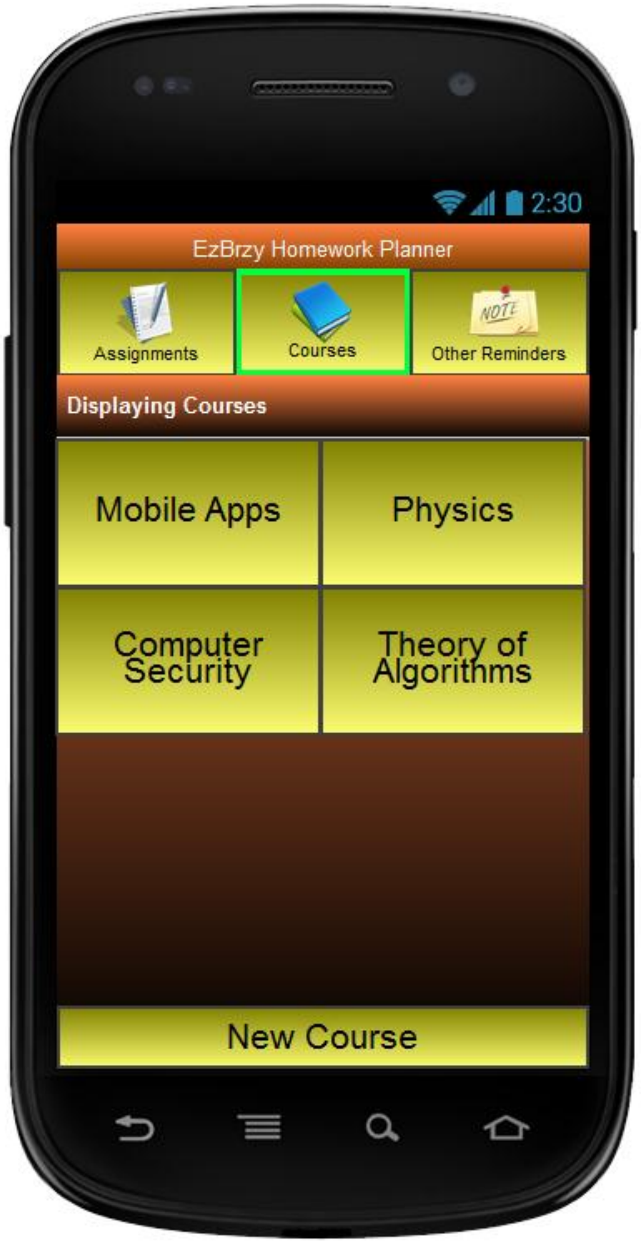
EzBrzy Homework Planner

Project Summary

Mobile Application: V.1.0



Summary by: Cora Charneskey & Eric Wilcox

Technologies Used

PhoneGap (Cordova)

- PhoneGap/Cordova was the chosen platform for building the EzBrzy Homework Planner. PhoneGap is open source, cross platform, and is based on JavaScript, HTML, and CSS. It provides support and documentation to several common API's that include Android, IOS, Windows 7, and BlackBerry. Setting up the PhoneGap environment can be slightly difficult, however a good tutorial on exactly what to do is provided by their site. Certain versions may work better than others, however toward the end of this project we decided to upgrade to Cordova 2.2 and it seemed to solve some of the bugs we were experiencing.

Android API Methods Used

- Since tables were needed to store the applications data locally, we chose to use the Storage API outlined by PhoneGap. The transfer of information between the database we have set up and the specific DOM elements has been the most difficult task so far. To begin, a database object must be declared and then we can perform transaction operations on the newly created database object which include creating or dropping tables, inserting data into the database, deleting database data, and calling queries.

jQueryMobile & HTML5

- jQueryMobile was used because of its mobile platform flexibility, common jQuery syntax, availability of the ThemeRoller application and since it works nicely with HTML5. The HTML5 attribute data-role="page" allowed all HTML code to live on one single html page since we could divide the sections into pages within one page by giving each page section an id. Other data-role's such has header, content, footer, button, listview and navbar simplified the styling process and integrated well with the custom theme created by the ThemeRoller application. The ThemeRoller allows you to set all the general styling properties without spending tons of time learning and writing css code.

Mobiscroll

- Mobiscroll was the plug-in chosen to implement the time, date, and occurrence scrollers that we have included in this application. The Mobilscroll API relies on the jQuery library which was already included in our application. The object shows up when a person chooses the time, date or occurrence field and this was easily done by placing a function within our javascript file that is called when the text area is clicked.

Work Done So Far

- Code the application skeleton (done by Eric and Cora, 10 hours)
- Forms for adding courses , assignments, and notes (done by Cora, 2 hours)
- Choose and implement custom icons (done by Eric, 4 hours)
- Write functions to display the number of items listed on each page (done by Eric, 6 hours)
- Install and implement the Mobiscroll plugin for date and time picker (done by Cora, 3 hours)
- Implement an organized file system between classes, assignment, and notes (done by Eric 60+ hours)
- Create theme through theme-roller (done by Eric and Cora, 4 hours)
- Implement alert dialog when deleting items (done by Cora, 2 hours)
- Link Save button to form submit (done by Eric, 5 hours)
- Add an "Occurrence" field to assignments and implement Mobiscroll (done by Cora, 1 hour)
- Need to clear form fields after adding the reminder (done by Cora, 2 hours)
- Bind the back buttons to history.back() (done by Eric, 1 hour)
- Documentation (done by Cora, 10 hours)

To Do List

- Discuss the DB setup and how to uninstall all the files associated with the device when a person uninstalls the app. If you do not do this, then when you re-install app files will still exist and conflict with each other. "Sqlite returned error code 14 cannot open file at source"
- Linking the native alarm API to application
- Make the reminder button function using native alarm
- Review and adjust device permission
- Validate data that is stored to local storage. Use the Jquery plug-in, jQuery.validate, we are looking for length and html characters. Also, remove special characters.

Work to Be Done for Version 2

- Implement native camera to store pictures related to assignments
- Work with native device calendar for additional display of recorded reminders
- Address minor bugs throughout the entire application

Modifications Due to Difficulties

- Chose to use a "listview" to display courses rather than the original 2 x 2 design
- Chose to list captions for form inputs above the text area because placing the captions to the left of the area was difficult and looked bad