

Classfinder 2.0

Geoff Shannon

Kyle Rader

Xiangyu Pi

Introduction

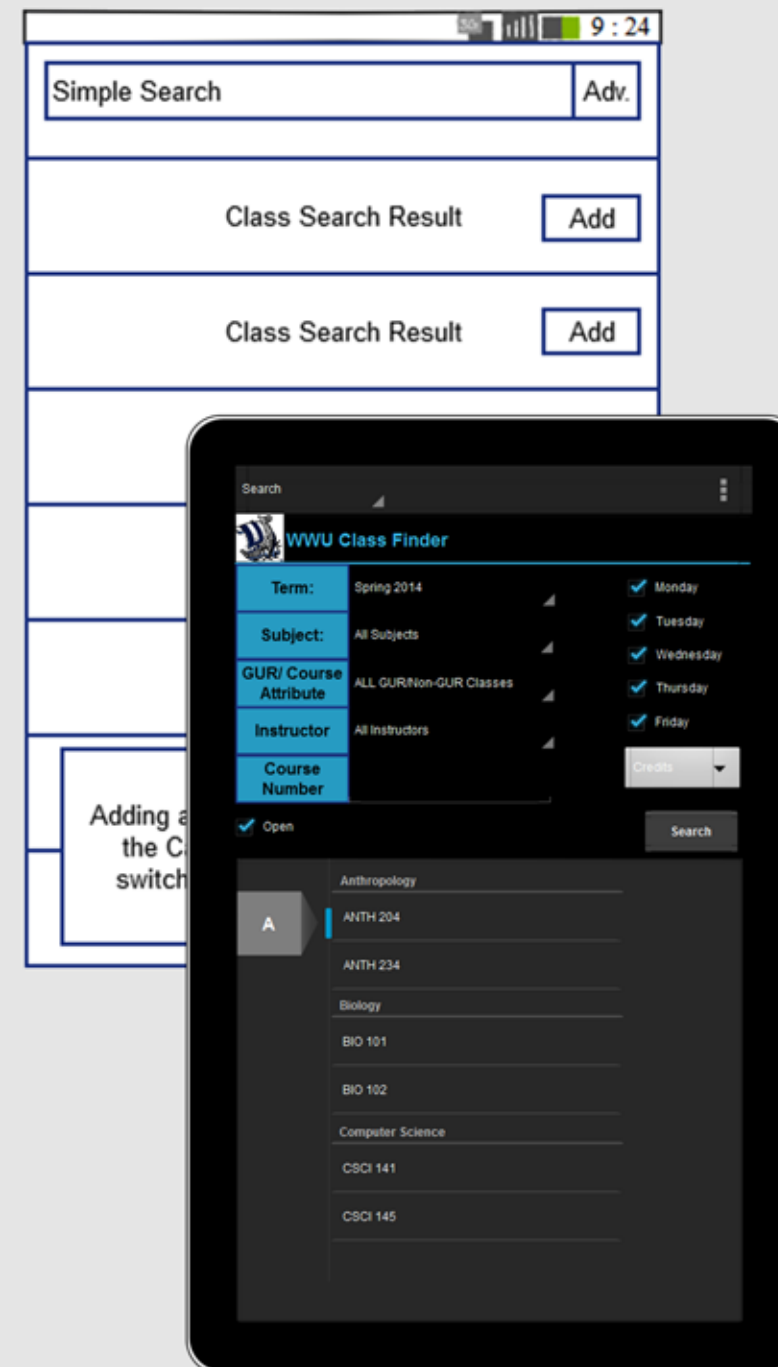
Classfinder 2.0 provides a mobile interface to Western's class registration system. Although the official Classfinder web application was once a technological masterpiece, it has not been updated significantly in the past decade.

On mobile devices, the application shows its age even more, making the site near-impossible to navigate.

The main goals for Classfinder 2.0 on Android are thus:

- Provide a modern user interface
- Provide a robust search interface
- Provide an easy-to-use scheduling interface

An Android application for interacting with Western Classfinder



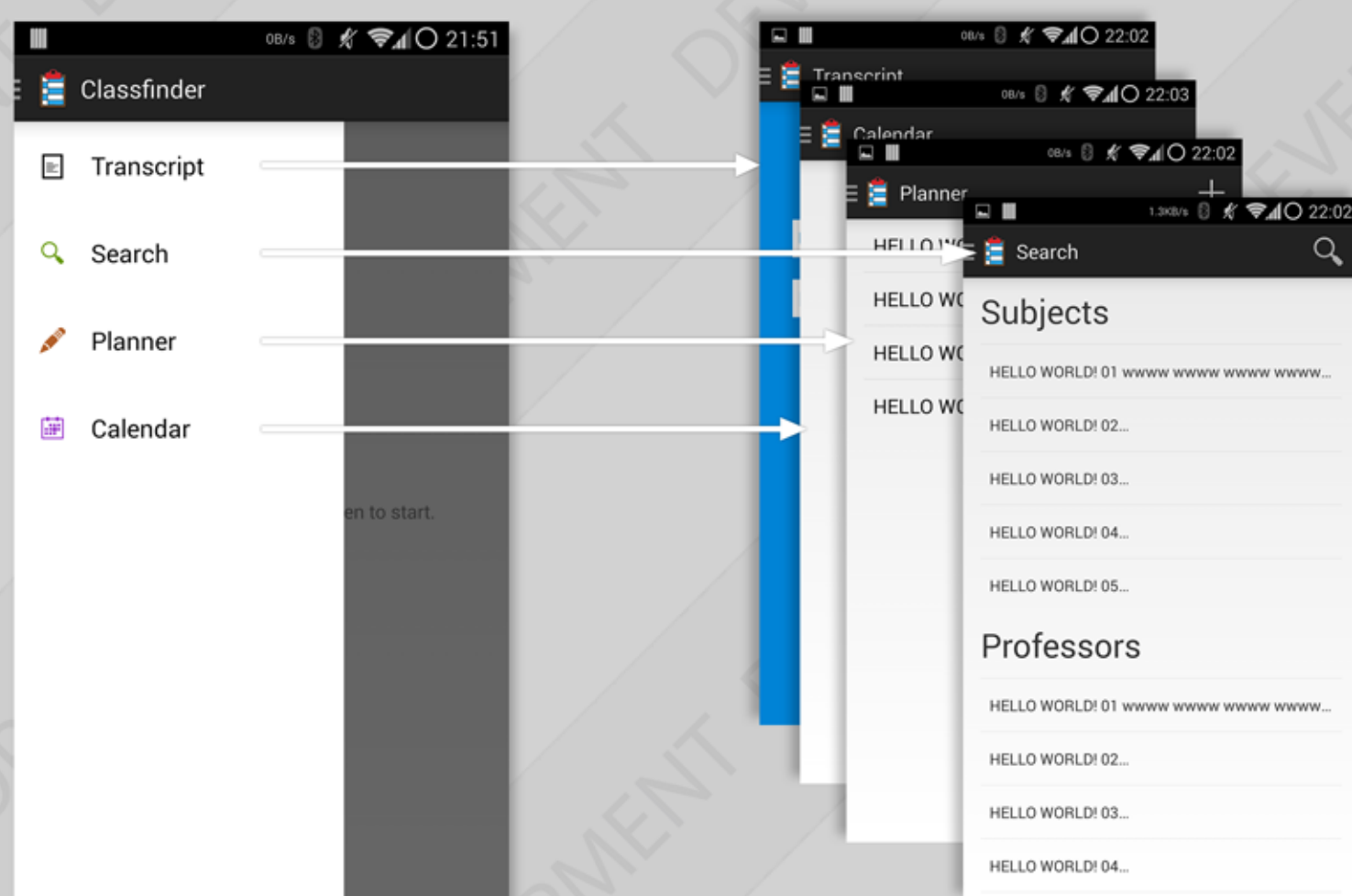
Pictured above: mockups of our early impressions of what the app was going to look like from the design documents.

In developing the app we opted to create our own API for classfinder. Having our own API gives a far greater deal of control than otherwise possible.

With our own API, we can request only content we are interested in. This optimizes cellular bandwidth and device storage, which also improves performance locally on the device by not having to retrieve (and search) through the entire list of classes, but only the relevant classes to a query.

This would not have been possible by simply scraping Classfinder, which would have resulted in a very sluggish application. This design also ensures that the application can be transparently moved over to the main Classfinder site, should they ever implement their own API.

The front-end



One Activity (Navigation Drawer)

Multiple fragments (and nested fragments)

One activity controls all the main views (fragments) in the application via a Navigation Drawer.

Each fragment then handles their own sub-fragments and interfaces with the back-end to display data.



Database

The back-end

The SyncAdapter periodically contacts the server for updated data, which are then cached locally.

Methods that return data are exposed for the front-end to display.



Android Phone



Server



SyncAdapter



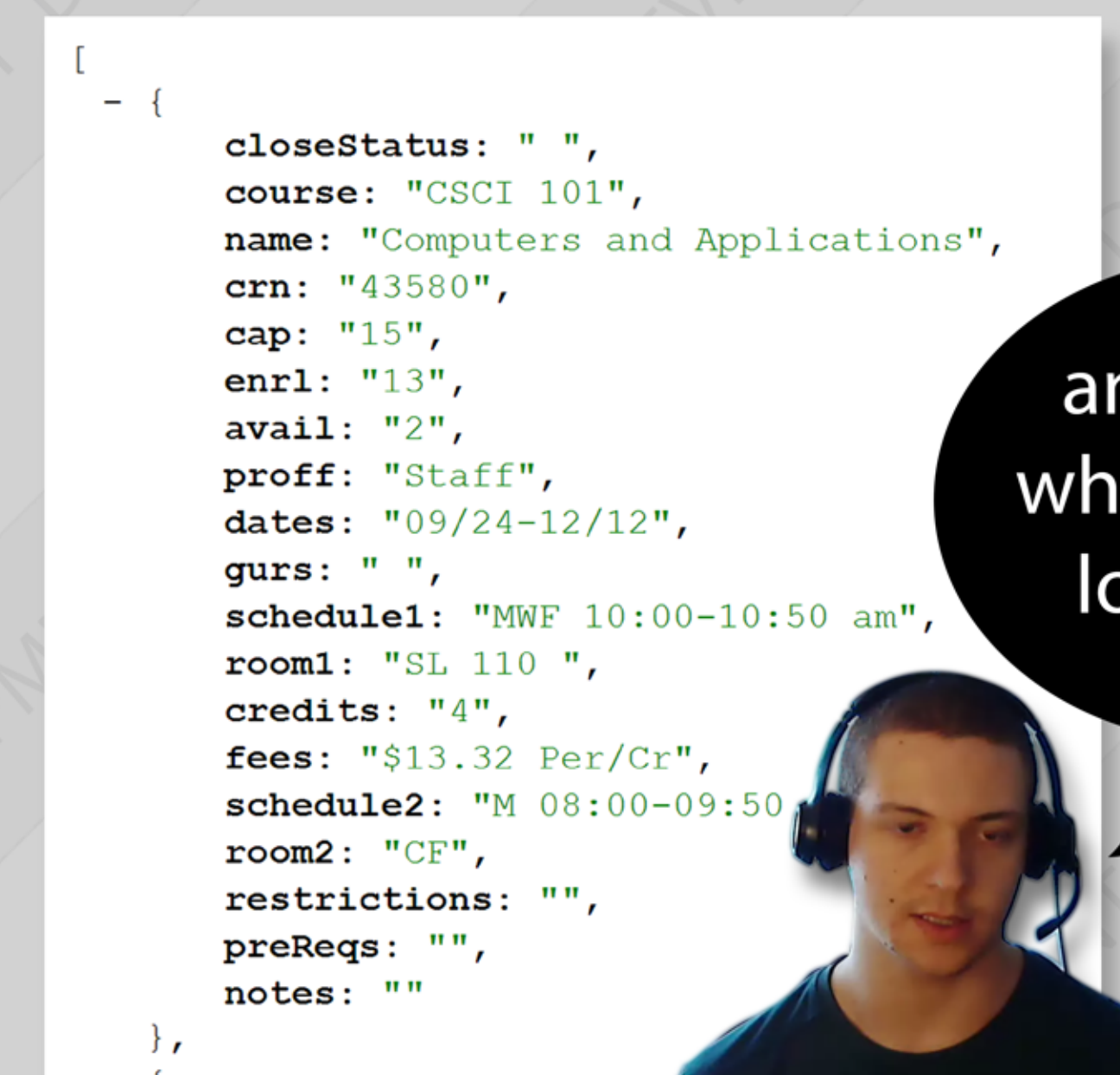
Database

The server-end

	M 08:00-09:50 am
CLOSED: Waitlist Available	CSCI 101
	Computers and Applications
	MWF 10:00-10:50 am
	M 12:00-01:50 pm
	CSCI 101
	Computers and Applications
	MWF 10:00-10:50 am
	T 08:00-09:50 am
CLOSED: Waitlist Available	CSCI 101
	Computers and Applications
	MWF 10:00-10:50 am
	T 12:00-01:50 pm

UGLY HTML

BEAUTIFUL JSON



For the front-end, we decided to use the navigation drawer paradigm as the main UI element. Our app has numerous views which are loosely related to each other. The navigation drawer UI helps to tie all these disparate parts together, letting the user quickly switch between views.

Since the navigation drawer only supports fragments, the entire app had to be written with fragments in mind. For more complex views like the planner and search views, care had to be taken on configuration changes and swapping out fragments to change the view.

Each fragment interfaces with the back-end to present data to the end-user.

The back-end is the glue between our front-end and server-end processes, ensuring the data is up-to-date from the server and presents that data to the front-end.

To ensure consistency between the databases between the phone and server, we use Android's SyncAdapter to periodically sync the data using our API. The back-end exposes methods for querying the data which the front-end makes use of.

Since the data from the server-end is cached locally, even when offline the app would be fully functional, limited only by the age of the cache and how comprehensive the cached data is.

Final Product

We made significant progress on this app. The front-end work is mostly done, requiring implementation of a custom view for schedule planning.

The back-end infrastructure is still not fully functional, and also needs to be hooked up to the front-end UI.

The server-side API provides a basic wrapper around Classfinder, however it still needs polishing.