Budding Budget

owsenk	Kyle Owsen
lizs3434	Elizabeth Schibig
ischaaf	Isaac Schaaf
jbktsang	Jessica Tsang
hstefan	Stefan Holdener
mjsc	Maxton Scott Coulson

Features and Cuts

Our vision for the app at the start was more ambitious than what we were able to accomplish. None of the stretch features we considered made in into the final project, and even some of the main functionality of the app had to be cut back on. Fortunately, our main use cases did get implemented.

The largest area of unexpected complexity was definitely the network and network integration. We didn't expect there to be so many issues, but a combination of the development teams for the app and server being separated, as well as issues inherent to writing any application that syncs data over a network led to a huge source of bugs and frustration.

Furthermore, most of the team were unfamiliar with the frameworks they were working in (and in some cases the frameworks were changed midway through production) which added to the time required.

Many of the features that were cut, such as notifications for recurring charges or diverting extra money to savings, were not expected to be as complex as they ended up being. Initially, they didn't seem any more complex than anything else we were doing with the same data - but it became clear as we tried to implement them that they would take significantly more work. As such, I think if we had about an extra week of development time, we could have finished all core features of the app.

Task Assignment

Elizabeth - UI

The most time-consuming part was learning and understanding the frameworks that were used. Having never used javascript or Jquery before required a fast learning curve to get the project done. One of the difficult things to do is UI development is to get the app to 'look good'. The CSS animations and other effects to achieve a fluid app is half the work required in UI development. A lot of time was also spent getting plugins to work correctly. One example is the JoyRide plugin used to create the in app tutorial. This was not the first plugin I looked at, but the only one I could get to work correctly with our code.

Overall this turned out to be more work and take up more time that I originally expected. I did not feel that I spent too little time on some feature in the UI.

Jessica - UI

My expectations lined up pretty well with front end development. We didn't plan the code structure well so we ended up with one large UIVew.js file that we had to later divide up into their own modules. Overall, we followed the spec pretty closely with the exception of inserting login into the program flow.

I spent most of my time learning how to effectively use jQuery Mobile and NativeDroid2. Debugging javascript took up the majority of my time since I am not too familiar with the language as well as the frameworks that were used. We spent too much time working with ChocolateChip-UI and ended up scrapping it. This was due to not spending enough time researching the framework.

Stefan - Algorithm

The algorithm was not as straightforward as expected. A basic version without incorporating recurring income and charges was quite simple but the full version required quite a bit of testing out different algorithms. Testing it revealed some edge cases that needed more attention.

Kyle - Phonegap, local storage, testing, project management

Phonegap ended up being a surprisingly small amount of the work in this category - it effectively just acted as a web browser, and there was very little extra work that we had to do with it. Local storage was the same way - once we settled on Mozilla's localforage library, it was relatively painless. Most of the time was spent on testing (and fixing related bugs from the tests), as well as project management. Synchronizing work and data representation between four teams took more time than it would have with a more seasoned manager, and it left me with less time to write tests than I wanted.

Isaac - Server

The server ended up being a fairly time-consuming entity. Overall testing and bug finding was the hardest thing to accomplish due to the large amount of code involved. I think where I probably spent more time than I needed was on validating input. Although making sure the input data is valid is important, much of it was redundant since the client also validated its data before sending it.

Maxton - Network, network integration

Writing code for a network was entirely new to me so my standard techniques of writing code and immediately testing were not sufficient for the tasks required. In the end I had involvement in the client side network as well as the login/creation for users, however since I struggled at times I received help from Kyle and Isaac which showed me some of the benefits of working in a team structure.

Detailed Schedule

Numbers in parentheses represent developer days.

Week - Goals	Front End	Algorithm	Phonegap	Network
Th : 1/28- Finish Software Design Spec	Divided up sections of the SDS document between members of the whole team, review final document together.			
T : 2/2- Finish Design Presentation Due Tuesday 2/2 Th: 2/4- Work on documentation for users and developers, start zero-feature release	Split requirements, design, and planning slides for presentation. Assign half of team to practice presenting.			
	Starting coding, designing the UI. Learning the toolchain. (3)	Start working on product website and user / developer documentation instead. (1)	Set up version control, bug tracking, other tools (2)	No back end work. Zero feature release is UI only. Assisted other teams
M: 2/8- Zero-feature release due Th: 2/11- Update SRS, SDS and other documentation, establish unit tests	Completely change UI Framework since original vanished.(4)	Look into potential ways to implement algorithm. (2)	Look into feasibility of Phonegap notifications. (2)	Get cloud database setup. (3)
	Begin updating documentation. (2)			
T: 2/16- Major features implemented	Get core features to the app (front end and back end) integrated. (3)			

Th: 2/18- Prep for demo. If time allowed: Started stretch features F: 2/19- Beta Release	Finish most of UI (8)	Continued work on calculator. (2)	Work on Notifications + local storage + testing (4)	Debug/clean existing code. (3)
	All prep for demo (2)			
T: 2/23- Work on stretch features Th: 2/25- Nearly all relevant bugs resolved. Update all documentation. F: 2/26- Feature-complete release due	Implementing hooks in the UI and local storage to integrate with backend code in the future. (6)	Recurring charges and income on the calculator. (2)	Find test coverage tool, continue writing tests. (4)	Connecting server to the client - implementing NetworkManager on the client side. (3)
T: 3/1- Perform/finish user testing Th: 3/3- Work on code reviews, all bugs resolved. F: 3/4- Release Candidate due	Refactoring UI code, looking into feasibility of adding tests. (1)	Calculator tests. (1)	Code review of UI Code. (3)	Fixing bugs in network and setting up account login and creation. (4)
	User testing (3)			
T: 3/8- Final Release W: 3/9- Final Presentation	All work on wrap-up and prep for presentations (3)			