



Akira Monri
Li Haoyi
Nahom Workie

Design

Record Page

Our record expenditure page is where the user lands upon opening our mobile website, and is the place where the user records down an expenditure that just occurred. This view includes an area for the user to enter the amount recorded, dropdowns to set the date of the expenditure (auto-filled to today) as well as the a dropdown to select what kind of expenditure it is. It also shows a map, to let the user know where the phone *thinks* the user currently is.

The screenshot displays the mobile application interface for recording an expenditure. The top status bar shows the time as 17:11 and various system icons. The browser address bar contains the URL 'http://23.21.59.23/'. The Traxx logo is prominently displayed in blue, with a 'Logout' button to its right. A navigation bar below the logo features four tabs: 'Record' (highlighted in blue), 'Transfer', 'Debts', and 'Analyze'. The main form area includes a text input field for the amount spent, currently showing '0.00', followed by a unit dropdown menu set to 'in'. Below the amount field is a dropdown menu for the expenditure type, currently set to 'Cash'. There are three date dropdown menus: '2012', 'May', and '15'. A map section shows a location pin and the text 'Tap to get location'. At the bottom, there are two buttons: 'Submit' and 'Add Debts'.

There is a button to associate debts to this record, as well as one to submit the record to the database.

The expected use case for this screen is immediately after the user has spent money. In this case, the only thing the user has to input is the amount: the date and location should already be correct, and the type of expenditure (cash) should be correct for the majority of cases.

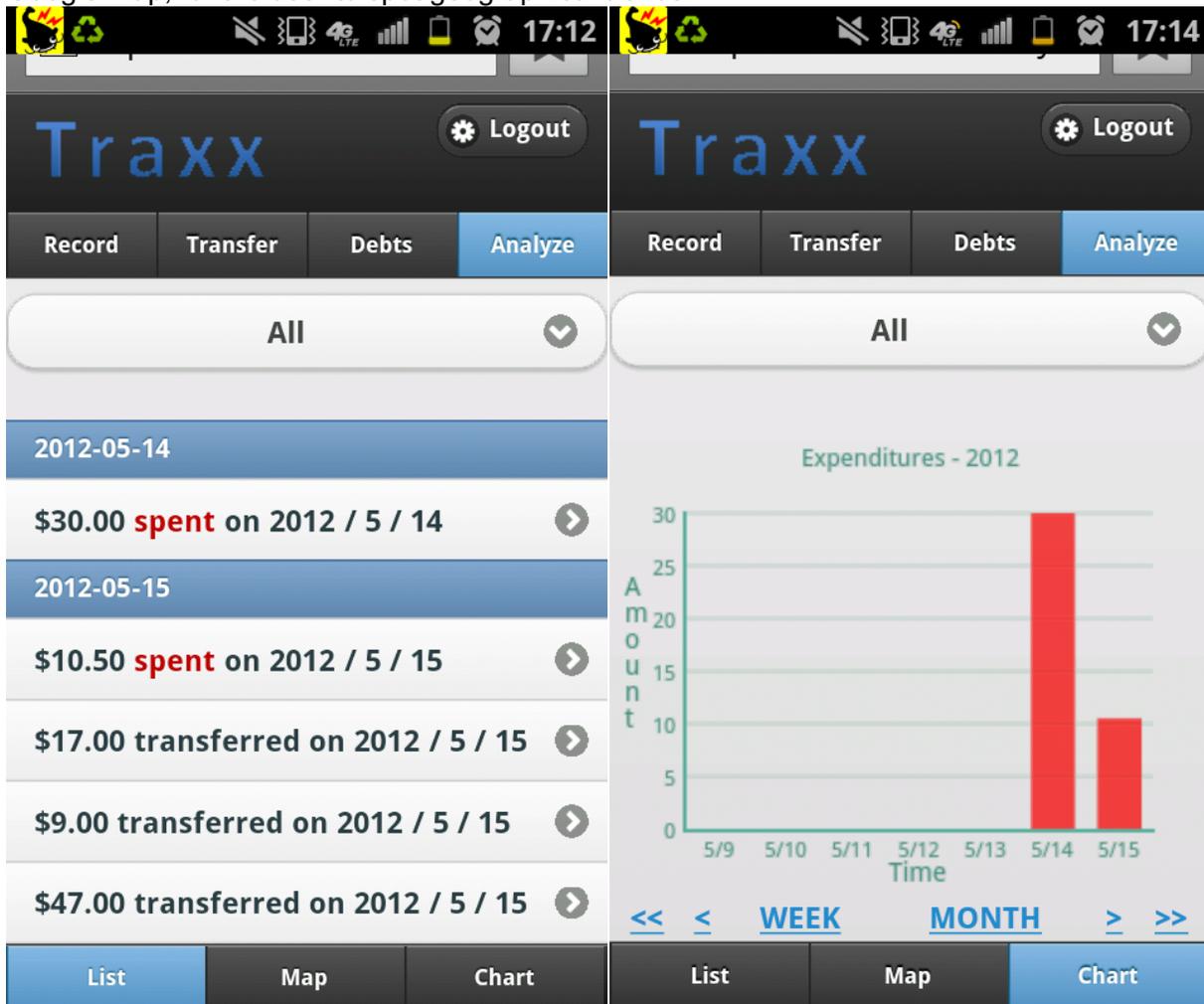
The Add Debts button is for the common case where an expenditure is shared by multiple people, and a single person pays for everyone else's expenditure. This allows the spender to easily jot down who owes who money, such that these debts can be resolved later.

The changes to this page include:

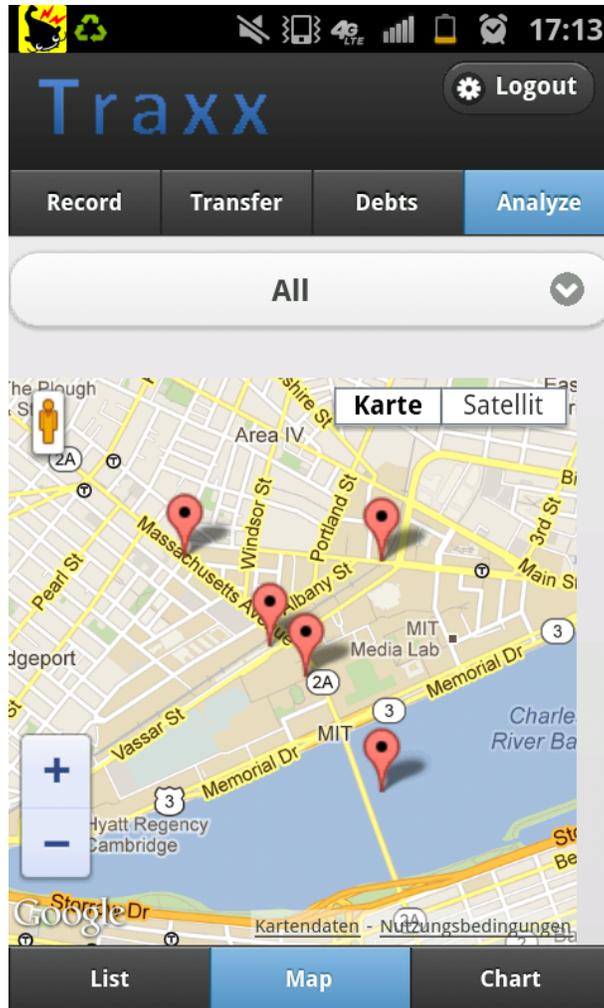
- Removal of the "Take Photo" method of recording the receipt; this was due to technical reasons, as the device camera is not fully supported across mobile browsers
- Modifying the wording into sentence-form: "I spent 0.00" instead of "amount spent: 0.00", as it is easier for first time users to understand when structured this way

Analytics

Our Analytics page has three views: a List view, which is a sorted list of all transactions that have occurred. A Chart view, which allows aggregate expenditures to be viewed over a period of a week or a month. A Map view, which plots out the positions of the various expenditures on a Google Map, for the user to spot geographical trends.



The List view is pretty straightforward. On the Chart view, we wanted to make the chart interactive, but were unable to find a mobile-compatible javascript charts library. As a result, the Chart is static.



The Maps view allows the user to see where he spends money. Clicking on any individual marker will bring you to the detail page for that record, allowing you to view the amount spent and other data associated with that record.

The major change to this set of pages was the separation of the three pages. Initially, the three pages were meant as different views of the same data-set: similar to how google analytics works, filtering by something on one page (e.g. by a certain area on the map) would filter the other views (list and chart) as well, when the user navigated there. We had thought that this sort of filtering would be incredibly powerful and easy to use, but it turned out to be difficult to understand, especially because we did not have space to add any breadcrumb trail to indicate what filters are currently being applied.

As a result, these pages are now completely separate, with different ways of being filtered and sorted. It is a loss of utility, but at a great gain in usability.

Add Debts/Transfers

The image shows two side-by-side screenshots of a mobile application interface. Both screenshots feature a status bar at the top with various icons and the time 17:12. The application header includes the 'Traxx' logo and a 'Logout' button. Below the header is a navigation bar with four tabs: 'Record', 'Transfer', 'Debts', and 'Analyze'. The left screenshot shows the 'Transfer' tab selected, with a form for recording a transfer. The right screenshot shows the 'Debts' tab selected, with a form for recording a debt.

Left Screenshot (Transfer Page):

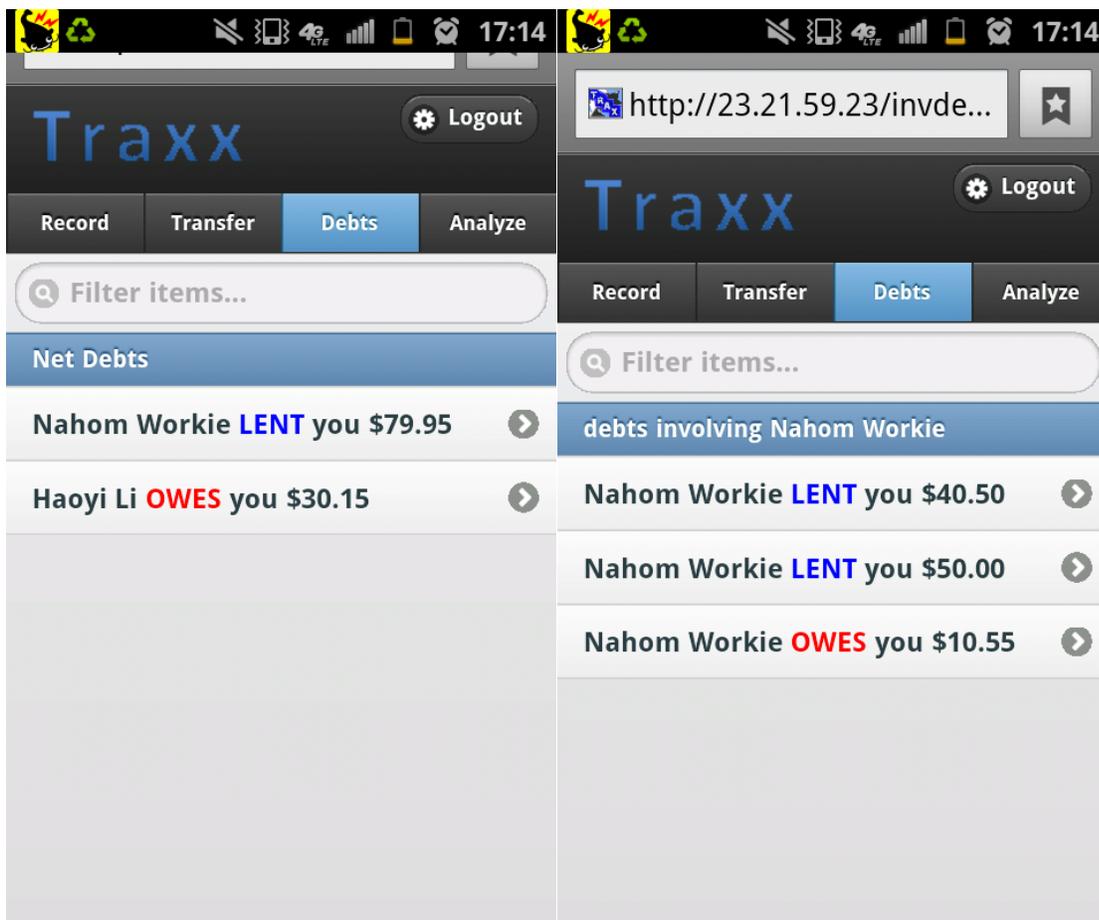
- URL: <http://23.21.59.23/transf...>
- Header: Traxx, Logout
- Navigation: Record, **Transfer**, Debts, Analyze
- Form Fields:
 - I Transfered \$: 60
 - From: Bank
 - To: Cash
 - Year: 2012
 - Month: 05
 - Day: 15
- Button: Transfer

Right Screenshot (Add Debt Page):

- URL: <http://23.21.59.23/addD...>
- Header: Traxx, Logout
- Navigation: Record, Transfer, **Debts**, Analyze
- Form Fields:
 - Name: Hao|
 - Relationship: Owes (dropdown), Me
 - Amount: 0.00
- Button: Done

These pages are relatively typical CRUD (Create, Read, Update, Delete) pages, and as such did not have a huge amount of design going into them. The one major change was to convert the helper text into sentence-format, similar to on the records page, so the user immediately knows what he has to do.

Debts Listing



The debts listing page allows the user to see all the debts involving him in the system, whether created by himself or by other people. Initially, it displays the net debts per person, and clicking on each person would bring you to a full listing of the individual debts related to that person.

We found that this was somewhat confusing, as the two views of net-debts-per-person and every-debt-relating-to-a-person look incredibly similar, but are in fact completely different, and people were indeed getting lost. As a result, we added a small list-view-divider at the top of the list indicating what the user was looking at (net debts, or debts relating to a person) in order to remove the ambiguity.

Implementation

Our application is a mobile web app, and the individual pages are simple HTML, generated by a web server, which also stores all data. Our stack comprises, from bottom up:

- Amazon EC2
- Nginx
- Flask
- JQuery Mobile

The actual logic is about 300 lines of Python, and compact enough that it fits in two files. For external API's, we used the Google Maps API to render all the pretty maps.

We do not have a database layer; all data is stored in memory as lists of Python objects, and disappears when the server is turned off. We currently have no security: no protection against CSRF for example, and no input validation before it goes into the data-set. These shortcomings are a conscious trade-off to focus more on the user interface

Evaluation

We conducted our user tests at two different places: one at a fraternity, and one at the student center on campus.

We found smartphone users with a variety of backgrounds and told them that this website is to keep track of their expenditures. They were not shown any demos. The tasks given were to:

1. Create your account
2. You just ate at a restaurant and paid 18 dollars as cash. You lend Akira/Haoyi/Nahom (Person who's talking to the tester) 8 dollars. Record your spending.
3. View my spending record as a map.
4. View my spending for the month.
5. Logout.

Results

Overall, we found that people really liked the Analytics portion of the app: being able to visualize spending geographically and temporally was very novel for many people, and they really enjoyed using it. The confusion that arose from our initial version of analytics, with the clever 3-way filter, was all gone.

The process of recording an expenditure was not bad - people generally had no problems with it - but could be improved. Possibilities for improvement would be a short NUX (New User Experience) that pops up the first time a new user visits the page, to help guide him through using the application and what he is meant to do. We are reluctant to put an explicit "How to use the App" landing page, as although it would increase initial learnability, having to repeatedly go to that page would limit efficiency in the long run

The process of adding a debt was universally confusing to people. This is something we did not expect, but we can see how it would be the case: adding a new debt requires navigating to a separate page, without any indication that it is related to the original record you were modifying. Furthermore, adding more than one debt requires taking you to yet *another* page: although this linked-list style of debt recording was convenient to implement, it is not a style of interface which users are generally used to. Ideally, we would fold the debts page into the record page, such that users would have immediate feedback that the debt they are adding is related to an individual record.

Individual Results

"C. B." - Course VI - Male - Droid user - Sophomore

1. Took a little bit to find out how to log-in
 - a. mostly didn't read the login screen until few seconds later
 - b. found out eventually on his own
2. Record process good
3. found map
4. found chart.
5. logged out successfully

Comments: Nice interface

“B. K.” - Course IX - Male - Droid user - Senior

1. Create account - login successful
2. Recording 18 dollars okay
3. Recording debt fail
 - a. used transfer.... got confused, but went with it anyway.
4. Viewing chart and map successful
5. logged out

Comments: The debts and transfers was a bit confusing.

“Y. D.” - Course III - Female - Senior

1. login went well
2. Record process went well
3. Got confused when creating debts and then having to resubmit under records
 - a. took a while to figure out the purpose of transfer
4. Viewed chart and map successfully
5. Logged out

General Comments:

- Very easy to use interface
- Really useful on a day-to-day basis
- Some features (map tracking) really enable the user to see where the user spent their money.

Suggestions:

- Add debits should be it's own tab, the process of recording a spending vs. recording a debt can sometime be confusing to the user.
- The transfer option is kind of confusing because you can't really transfer cash to cash, some type of lock their would make it a bit simpler for the user.

“R. A.” - Course VI - Male - Grad

1. Login went well
2. Recording process started slow but went well
3. Transferred \$20 to cash from card

4. Map and chart view successfully
5. logged out

General Comments:

- Attractive interface allows users to easily learn the features
- The multiple visualizations in Analyze allow the user to see their spending in multiple ways which is something that I really like about this app.
- I like the fact that the app takes into account and keeps track when you switch from debit/credit to cash.

Suggestions:

- Add debits should be its own tab, it's really confusing when I have to submit twice in order to keep track of various debts.
- I don't like the fact that the records screen is the first screen that you end up seeing. It gets really confusing as first page you get to see on this site because it gives no background or info on what you're actually supposed to do.

Reflection

Akira

I had a lot of fun making this website. I had never used jQuery Mobile, which handles things very nicely for the most part. It was also interesting to see the different problems we found when opening our site on the computer and a smartphone: some codes that worked on the computer crashed on the smartphone, or the size of the smartphone screen was much smaller than we had imagined. When having our prototype tested, I thought it was really interesting to see how much people who use the application for the first time get confused with what the makers thought was easy to understand. Ideally, I would want to continue working on this site and make the changes that reflect on the comments we received from our testers.

Nahom

I really enjoyed working on this website, I especially enjoyed working on the front end aspect of the site. The biggest difference was probably working on a mobile site rather than an actual one mainly because you have to consider and prioritize different things such as screen size which play critical roles when you transition from desktops to mobile devices. Overall, I think one of the things I would of tried to understand/work on more would be figuring out what features we can add and implement into the site using Flask. This was my first experience with the Flask framework, so I think given more time it would be one of the things that I would of tried to understand more and add to our application.

Haoyi

We chose to do a mobile web app mainly because it was something none of us had done before, so it was new and novel: normal websites are well-studied, and so are native mobile apps. I think that it was a very interesting experience, given that I was in parallel developing a native Android application for 21w.789. The parallels between the two techniques (the apps look more or less the same) as well as the contrasts between them (native Android is *much* more painful!).

It was also the first time I had worked in a group setting with multiple people doing somewhat homogeneous tasks: previous group projects always had a very clear separation of labour: one person would focus on one section of the project, and thus there was not a lot of task assigning or delegation to do, as everyone already had their work cut out for them. In this case, we were much more interchangeable, with any of us working on any part of the app.