Design Issues:

Architectural Issues:

1. Platform and User Interface Implementation

Choices: iOS 7.0.4 or Android 4.4  
Decision: Android 4.4

-User base: The Android operating system powers over 1 billion smartphones and tablets. This enables us to tap into an incredibly large market, which can lead us to making a greater impact with our application.

-Well-documented API: Android is an open-source software stack, which connects us as developers to a huge community. As this is the first time our team members are developing for mobile, we need a well-documented API to help us.

-Third party technologies; Android also gives us the freedom to use third-party tools for development, thus leaving a lot of room for innovation without affecting the usability aspect.

-Developing Restrictions: Developing for the iOS operating system requires the developer to use XCode, which is only available on the Mac OSX. Not all of our teammates own a Macintosh machine, therefore Android was the clear choice winner.

1. Server Hosting and Management

Choices: Heroku or Parse

Decision: Heroku

-Cloud: Heroku is a cloud application platform, which made it appealing to us as we can host our application on the cloud rather than host it from a physical Linux box.

-Speed, Security and Scalability: Heroku offers one of the fastest instances for servers, and this is one of our major criteria, as we need our app to be very responsive. Security is one of our concerns, as we will handle information that could be personal to our users. As our application is centered on impact, scalability is a big issue and Heroku offers a great backend to easily scale our application as it grows.

-Third-party Add-ons: Heroku offers a vast variety of services created by known third party developers, where as Parse does not. For example, Heroku offers database solutions for developers who are comfortable with SQL, NoSQL, etc., while Parse has just one backend database solution.

1. Database Hosting

Choices: PostgreSQL or MongoDB

Decision: PostgreSQL

-Relational Database(RDBMS): PostgreSQL is a powerful open-source object-relational database management system. Our application will be using a lot of the features of a relational database as we will have connections across our records, such as users connecting to deeds which then eventually connect to movements.

-Well-documented API: PostgreSQL is an open-source software, and the community behind is huge. For a lot of our team members, this is the first time using a database to manage object-based records therefore we needed a service that was well-documented and easy to use.

-SQL: Our teammates are more comfortable using a relational database management system rather than a NoSQL database management system. PostgreSQL was the clear choice here.

User Interaction issues:

1. Deeds

Choices: Custom or Pre-defined

Decision: Custom(but offers pre-defined suggestions as well)

Autrui is based on people creating an impact through their acts of kindness; therefore the act of recording a deed has to be very easy. Users will be able to connect to other users through these deeds. Deeds will consist of simple text descriptions. Our team settled on letting the users create custom deeds, but there will also be a database filled with suggestions from us, to help users get started and possible provide them inspiration.

1. Response time

Choices: To cache data or not to cache data

Decision: Cache most relevant data

To maintain quick response time, there will be efficient search algorithms at use to help users quickly connect to other users. These deeds will be parsed into the database with the aim of keeping data usage to a minimum. We will also cache the users most recent deeds and movements that they have been added, thus helping reduce response times.

1. User Accounts

Choices: Pre-exisiting Social networks or App users

Decision: Pre-exisiting social networks

Users will be asked to link their Facebook/Twitter accounts when they first sign up to Autrui. The application will parse the necessary data retrieved from the aforementioned social networks, and keep only what is required, such as the profile picture, friends, etc., thus keeping the data to a minimum. The reason we force users to use their existing social networks is so that they find it easier to find their friends more quickly, and also using the vast user base of an already existing network.

1. Movement Creation

Choices: Automatic or User enabled

Decision: Automatic

Movements will be created automatically when at least 5 people have been affected by an initial deed. We chose to do this as we didn’t want small movements of just 2 people being created. Users will be able to view these movements once their created through the app, with just a click of a button.

User Interface Issues

1. Color Scheme

Choices: Warm colors or Cool colors

Decision: Warm

Our team decided to go with more warm color palette as we wanted our app to be as inviting as possible. Granted we can achieve this effect using cool colors such as blues and greens, but the decision was unanimous amongst our team members.

1. User Stories

Choices: Newsfeed or no Newsfeed

Decision: No Newsfeed(but still leaving it open for further iterations)

Our team decided not to add a newsfeed for the initial version as we were going for a very simple, minimalistic look for Autrui. The addition is of a newsfeed does have its benefits, but the added clutter could affect our minimalistic touch.