CS 307

Project Title: Autrui

Team 10

Mihir Jham

Rishabh Mittal

Karan Kalwani

Manmohit Sehgal

Ankit Kapur

Design Document

A. Purpose

With the advent of technology in the daily lives of people in the past decade, it has been through many forms of uses – from being a luxury to utility to now a means of entertainment. We see that with though this transition has bought greater connectivity and access to information, it has also lost compassion somewhere down the road. The whole world of the social network seems cold and lacks benevolence. We have decided to carry the idea of bridging this gap, by creating a mobile application that aims at bridging this gap.

The app is inspired by the Pay It Forward Movement. The idea is to respond to a person's kindness to oneself by being kind to someone else. Our app focuses on providing a social platform where users will be able to connect through these kind deeds, and see the impact of their deed.

B. General Priorities

Keeping the purpose and the users of our application in mind we made a few decisions regarding the non-functional requirements. The decisions are made in order to guide us during the design process, by being aware of the final product and the direct user at all times.

- **Usability:** Our application is made with keeping in mind that almost anyone can be the end user of it. The user should be able to access information in the form of deeds he/she has created the impact of these deeds, the global footprint of these deeds and the deeds the user has been a part of and needs to proceed forward with. This will be achieved with efficiently organizing our work and keeping in mind the importance of the feedbacks.

- **Reliability:** Autrui will be based on the Android platform. The database will be created using PostgreSQL. The app will be hosted on Heroku (a cloud-based server management platform as a service (PaaS) supporting several programming languages). Amazon S3 will be used to store all the data. All these resources provide extremely reliable services. Making use of Android as the platform provides us with operational integrity. An applications performance is entirely dependent on the way it access the databases, pulls data from the servers. For Autrui, this becomes even more important as essentially a ‘movement’ can have n number of users linked to each other with an equally gargantuan number of deeds. Hence our choice of the right service is imperative to maintain the reliability of our app.

- **Accessibility:** The app and its services will only be available to the user via an internet connection/GPRS capability. The application will not be available offline for the user to make any changes.

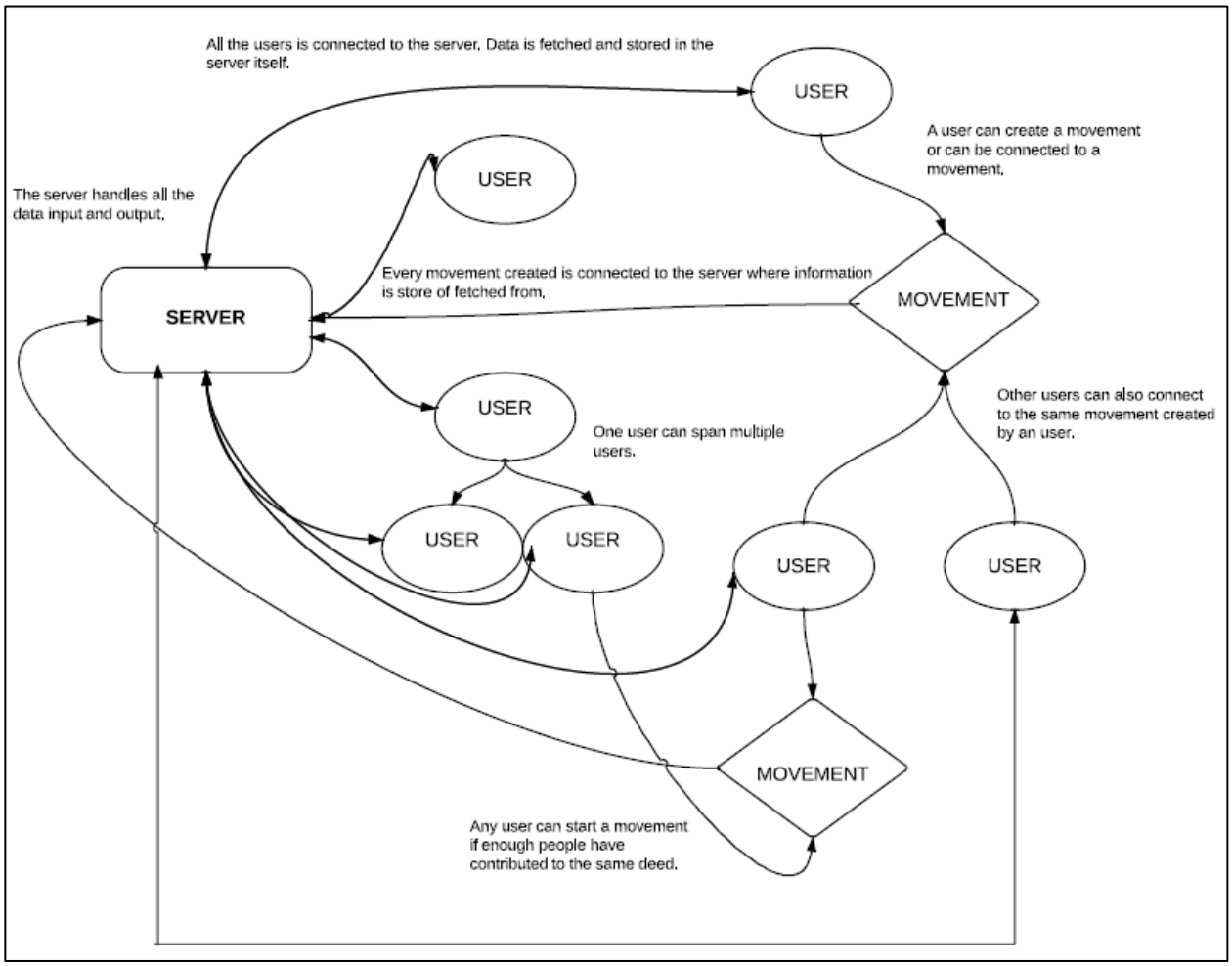
- **Performance:** In order to give the right shape to our idea and to truly exemplify the beauty of our app, it will be imperative to maintain top notch performance. Hence, having extremely efficient and robust algorithms and maintaining international coding standards will be our top priority.

- **Platform:** As part of our first sprint, we will be developing Autrui for the Android 4.4 framework. We do aim at making the app backward compatible which will be made extremely easy by making use of the appropriate libraries that are already provided on the Android SDK.

- **Stability:** As stated earlier, we are making use of resources that are already successful in the industry, making them an extremely stable, well supported with resources, tried and tested means.

C. Outline of the Design:

As shown in the design outline below, the structure of the application will be such that the users will be connected to each other by a deed. The deed may be a new one or may be a part of a larger movement. A movement will have at least five users connected to each other, directly or indirectly. Once a user creates a new deed, information regarding the initiator, the recipient and the deed itself is exchanged with the server. The recipient is notified by the server and is required to do two more deeds while being a part of this chain. Being of an incomplete deed chain does not restrict a user from creating a new deed in isolation.



D. Major 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.

2. *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.

3. *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.

2) 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.

3) *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.

4) *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.

2) *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.

E. Details of the design:

- **UI Mock-ups:**

Below, you can find the mock-ups of the user-interface.

1. The app will be introduced to the first-time user with a screen that provides them log-in options both via Facebook and Twitter. The screenshot does not feature the in-app registration and login option as we are still working on the design and the overall look of the same.

2. Once the user has successfully logged into the app, a host of options will be shown to perform specialized functions. Tapping the gear icon on the top left will take the user to the settings page which will provision various options to edit settings, log out, help etc. The peoples icon on the top right will enable the user to witness the global footprint of the deeds to see a representation of the global footprint of the deeds he/she has been a part of. This will make use of geo locations to provide exact points on a world map. Tapping on 'create a deed' will take the user to screenshot 3. Tapping on 'my movements' will show the user a list of all the movements he/she has been a part of till now (shown in screen 4).

3. Once the user taps the 'create a deed' option, he/she will be prompted to enter a short message explaining what the deed was. The user can then either tap the check mark to submit or the cross on the top left to discard the same.

4. On tapping the 'my movements' on screen 2, the user will be shown a list of all the movements that he/she has been a part of.

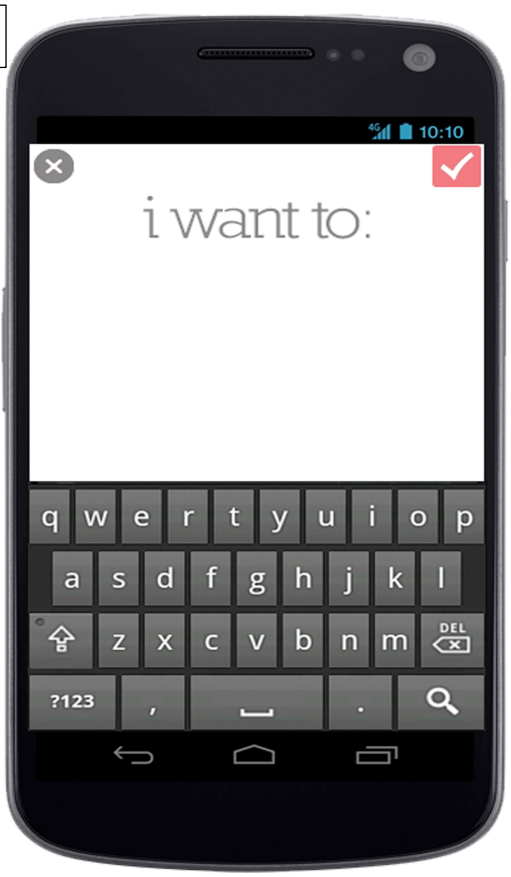
1.


11:02
autrui.
Sign-in with Facebook
Sign-in with Twitter

2.


4:20
autrui.
create a deed
my movements

3.

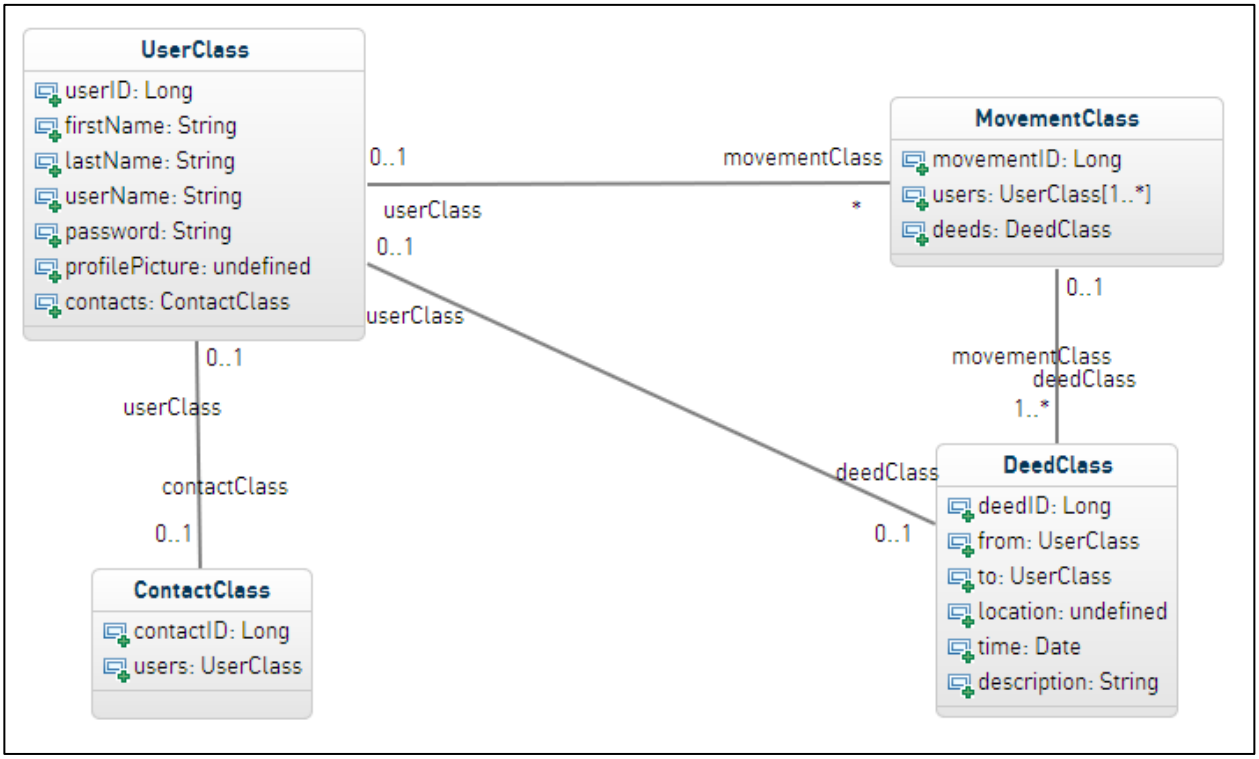

10:10
i want to:

4.


11:02
movement
10681 deeds | 594 people

- **Class Diagrams:**

You can find the class diagrams below defining the classes that we will be creating to make Autrui.



1. *UserClass:* This contains all the information about the user. Along with generic details like firstName, lastName, userName, password and profilePicture, we also userID to provide a unique ID to every user solely for development purposes – the user has nothing to do with this. We have contacts that essentially store the contacts that have been pulled out of the said user's Facebook friends list (after the user has approved us of doing so, of course). This class is connected to the other three classes – ContactsClass, MovementClass and DeedClass.

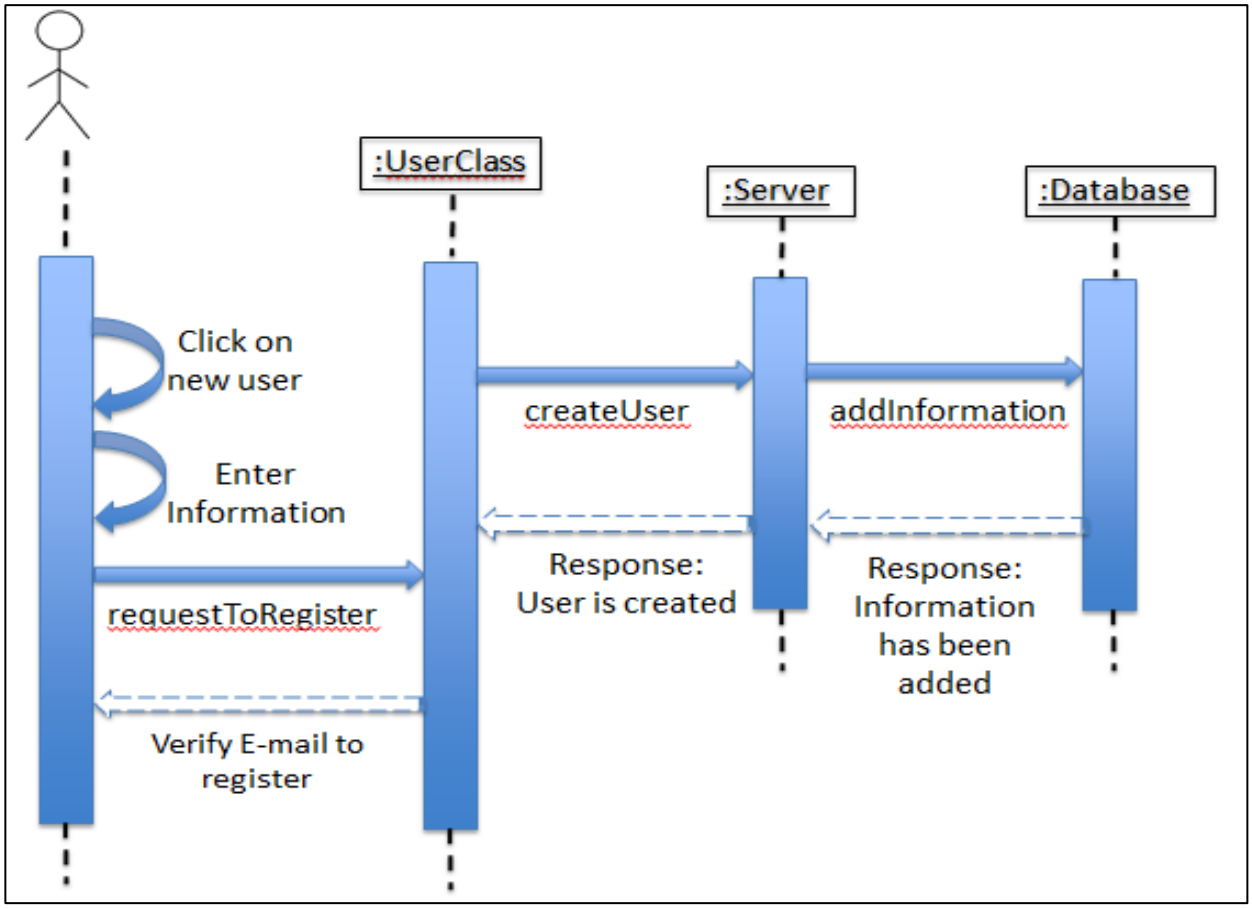
2. *ContactsClass:* This contains nothing but an array list of all the friends that the user has on his Facebook account, as described earlier. It also consists of the contactID that is only present for development requirements. This class is only connected to the UserClass.

3. *MovementClass:* This class consists of the movementID (to uniquely identify each movement), users (array of all the users that are part of a movement) and deeds (array of all the deeds that are connecting all the users in the said movement). A user can be part of many movements and many deeds can be part of a movement.

4. *DeedClass:* This class consists of the deedID (to uniquely identify each deed), from (the initiator of the deed), to (the recipient of the deed), location (geographical location where the deed took place – the recipients location), time (the date and time when the deed occurred) and description (to provide a small memo on what the deed is). A user can have many deeds and many deeds can be part of a movement.

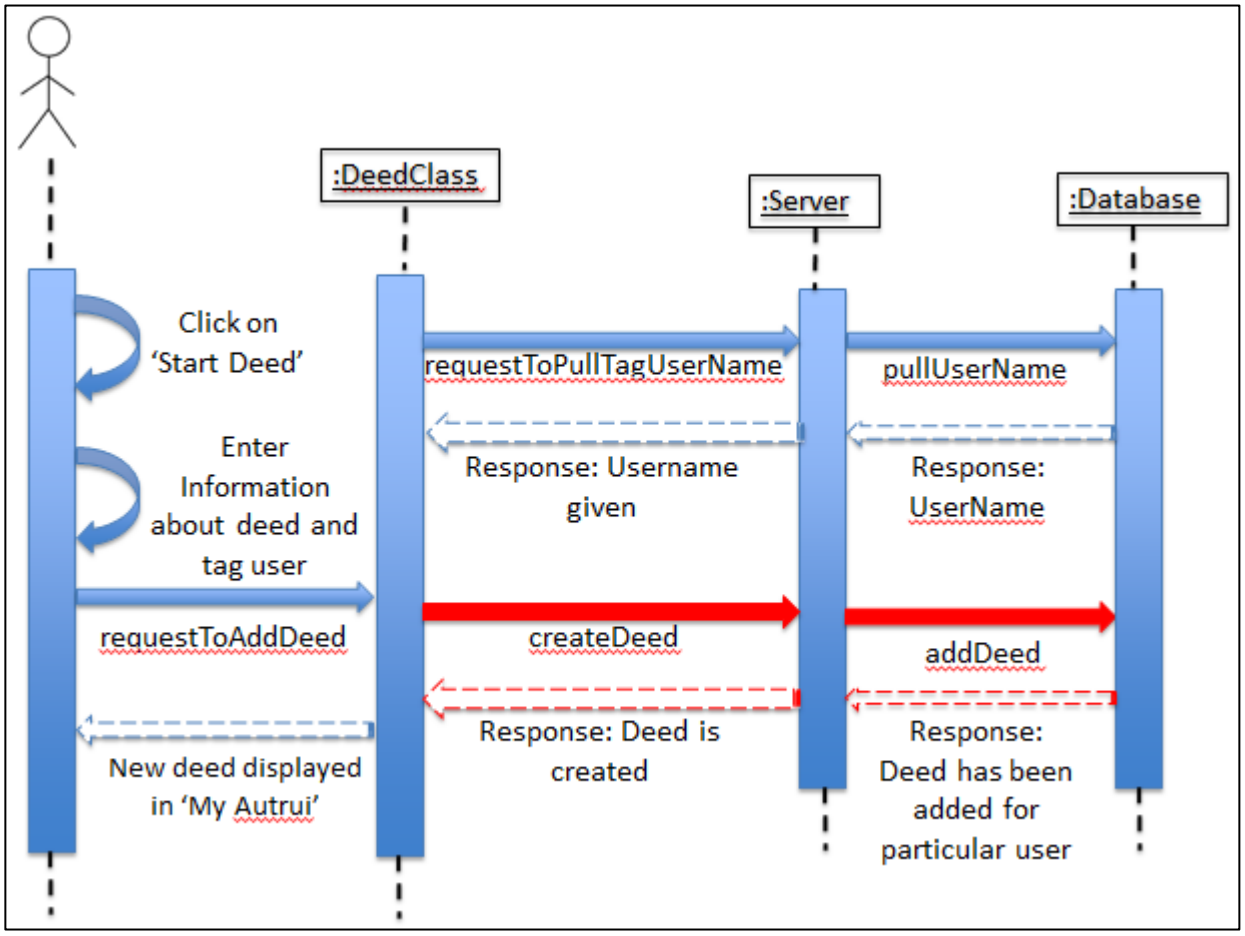
- **Sequence Diagrams:**

1. *User – Application E-mail Registration Interaction:*



This diagram illustrates the process which takes place when the user wants to register to the application. The user fills out the information and sends a request to be registered. For every user there is a object inside the UserClass which stores their information. The new user information is sent to the server and then the server adds the information to the database. The database responds back to the server with the response that the information has been added successfully. Then the server responds back to the UserClass with the response that the user has been created and only E-mail verification is required. After the user has verified their E-mail, they can view their information in the My Autrui.

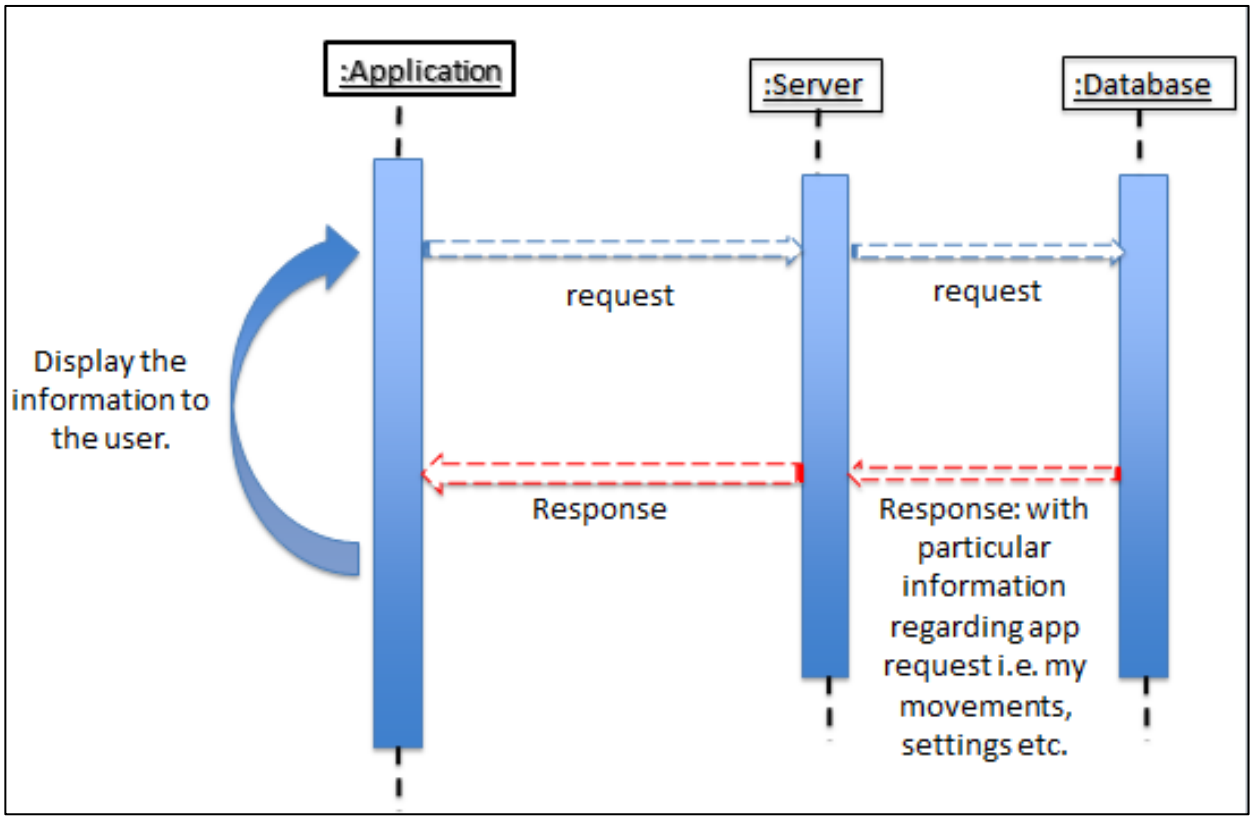
2. *User – Application Deed Creation Interaction:*



This diagram illustrates the process which takes place when the user wants to start a deed. The user fills out the necessary information i.e. a short description about the deed and tags the user for which the deed had been

done and then taps on the option 'Share'. At run time when the user wants to tag a user by typing in the name of the user, the deed class sends a request to the server, which in turn sends a request to the database to fetch the user's name and login id. When the user for whom the deed was performed, has been tagged and the user clicks on 'Share', it sends a request to be add the deed in the deed class for that particular user. The deed class then further sends to the server to create a deed. The server then adds the deed into the database and when the deed has been added to the database, the server refreshes the My Autrui page to show the latest content for that user.

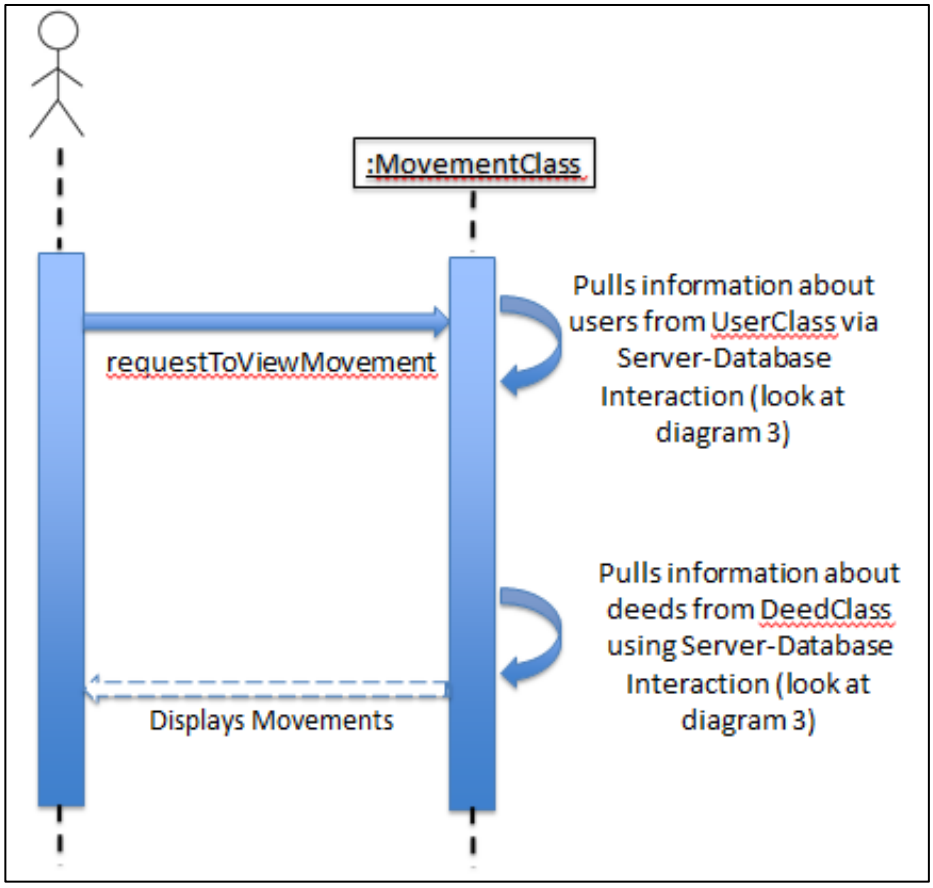
3. *Server – Database Application Interaction:*



This diagram essentially describes the interaction between the server and the database for any given action that the user might make. For example, say that the user may want to view the movements that he/she is part of. The user taps on 'my movements'. The application sends a request to the server which in turn sends a request to the database. The database responds to the server with the required information of the deeds and users connected in all

the movements that the user is a part of. The server sends this information to the app and this displays the information to the user.

4. *User – view movement:*



This diagram further explains the example presented in the description above. The user basically sends a request to view the movements. The movement class responds by doing the actions shown in diagram 3 and responds with the required information back to the user.