Design Decisions

- First decision is that we are using Git as version control system instead of using mercurial, vsn, etc. and GitHub as source code management instead of using Bitbucket, this decision was imposed by the teachers of the subject.

- We are using javascript family language(javascript, typescript, node.js, angular, etc) to build the application as the main programming language. As we understand, javascript is the easiest way to make a SOLID application.

- We are also using basic languages for the web such as html, css, etc.

- Social linked data(SOLID) based application. A must decision we have to follow if we want to make a decentralized application. This means following several important constraints to build the application. One of them is storing date in PODS, so the user is the owner of their data.

- Using a Service architecture that helps us to easily separate responsabilities. This services are used from the main app to obtain extra-functionality.

- Following Angular components structure, so we can build a high cohesion and low coupling app. Our main component is FriendComponent and it represents the chat page where we can select chats and send messages to our Solid friends. We decided to have all chat functionality in the same page because we though it was better for the final user as it will be able to move in our app with just a few clicks.

- We use the known Karma-Jasmine framework for implementing unit test, a tool that allow us to easily test code on Angular, it helps us separating test code from the application code, it is also useful and easy to learn how to test. We though about using Mocha.js for tests but we had a few tests done yet so we decide to keep on with Karma-Jasmine.

- At the beginning we used a simple text file to keep message info. Later on we decide to store messages in a ttl file. We stored message content and metadata in Turtle language (.ttl). The main reason to store data in rdf is that we can communicate with other applications that have the same ontology.

- As we store message data in ttl format, we need a tool for getting that ttl info. The way we recover that info is with Sparql queries, We create queries to access the Turtle files and get resources faster than using plain text.

- We have a notifications service that works in every browser and allows us to send a notification to a user when he receives a message. We decide to use this notification service because we think it's relevant in a real time chat application.

- Group chat develop is in a alpha version, we have decided to store all the conversation in group creators/owner pod.