1. Project Title and Authors

Team No. 52
Project Title and Authors
Carlos Subramanian Vidal (cksubram@usc.edu),Christian Addei (caddei@usc.edu), Henry Sazo (hsazo@usc.edu)

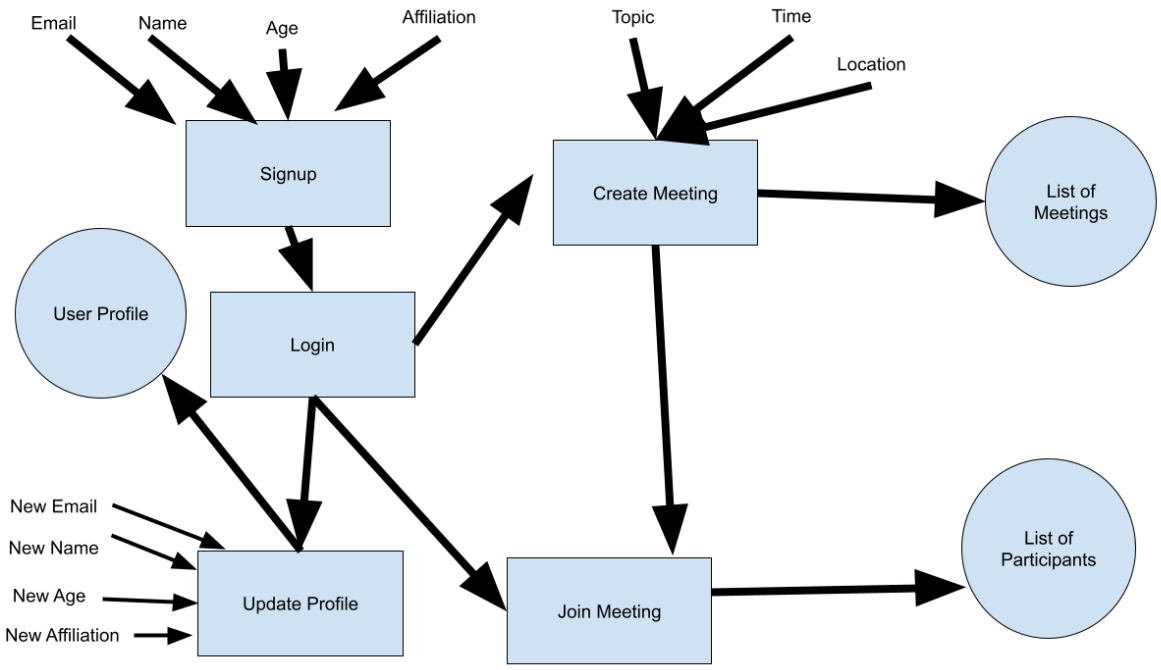
2. Preface

This document is for the technical professionals to see and understand how we came up with our implementation. Below you will find an updated architectural design and detailed design. You will also find some of the changes that we made to our implementation of certain features and the changes that caused in our detailed design. Ultimately, how it shaped our application to be what it is and how it works.

3. Introduction

This document describes below the updated architectural design and detailed designs we used for our implementation of the Talk2Friends application. From our previous duocmuent, we made massive changes to the detailed design by adding more classes, more member variables and member functions, and relying more on the use of a database for storage on the cloud rather than taking up local storage on our systems.

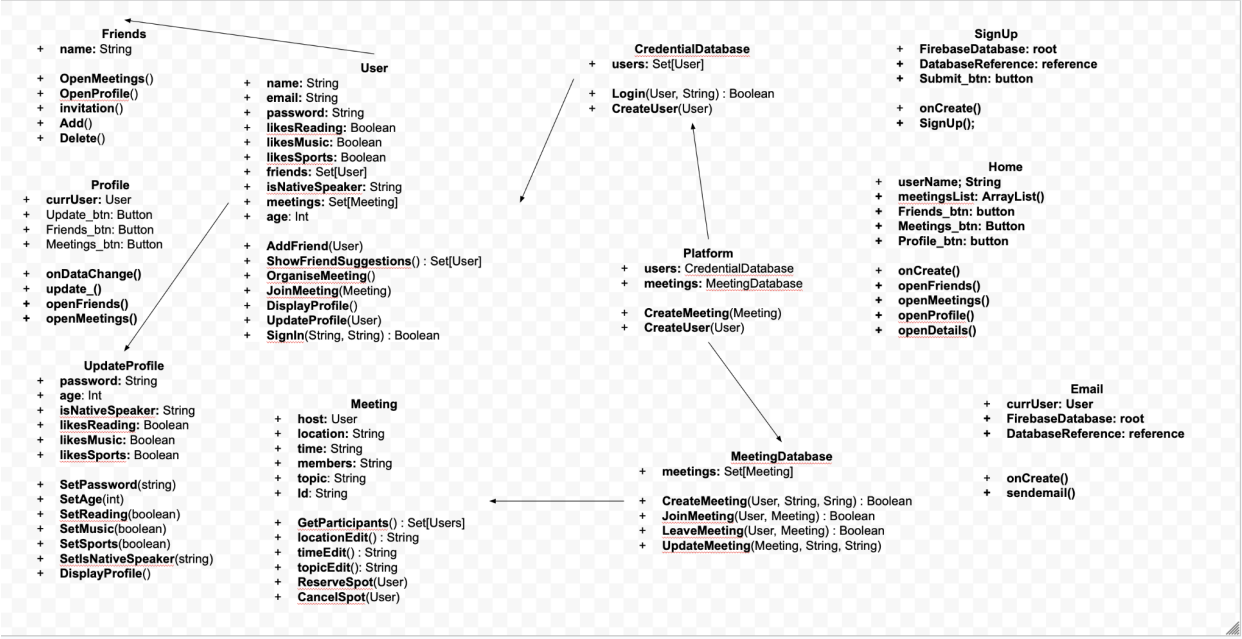
4. Architectural Design (Pipe and Filter)



Components: Signup, Login, Create Meeting, Join Meeting

The components represent important functions used in the app. The pipe and filter architectural style was chosen to show the stream of data through those functions in a simple display. For the most part this is what we still used. Structurally, this stayed the same when we implemented it in our program.

5. Detailed Design



Changes: From our previous documentation of the Detailed Design, we implemented more classes and files into our project. For instance, we made a separate class for “Email” “home” and “Sign Up”. We did this in order to keep the code clean from smashing together a lot of different processes into one or two classes like we had originally outlined in our previous detailed design. We also added many more member variables and functions to each and every single one of the main implementation classes. We found that by doing this, we can store more into the database and make it much more efficient in terms of memory if we depended on the database to do more information retrieval instead of trying to store it locally in our system.

6. Requirement Changes

We made a few changes to the requirement that only USC affiliate people can join the app. When signing up we put a restriction that doesn’t allow people with any different @----- for a gmail to join. As well as, when we invite friends, only people with @usc.edu are allowed to send an invite. IF not then the system wont allow them to send an invitation. And this does change our design, because we would have to add a validator function to make sure we are not letting non-usc-affiliate people join.

Another change we did was, when a user is signing up, that is when we ask them for interests that are later used to recommend friends to invite. This changed our design in the sense that from the very beginning we ask the user to check off interests and immediately store them on the database. So then we don’t have to run that later when they have to recommend friends. THis made our system more efficient and clean as we got it done before the code became jumbled with other features found in the friends tab of the application.