1. Project Title and Authors **1**
2. Preface **1**
3. Introduction **1**
4. Architectural Design (Pipe and Filter) **2**
5. Detailed Design **2**

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)

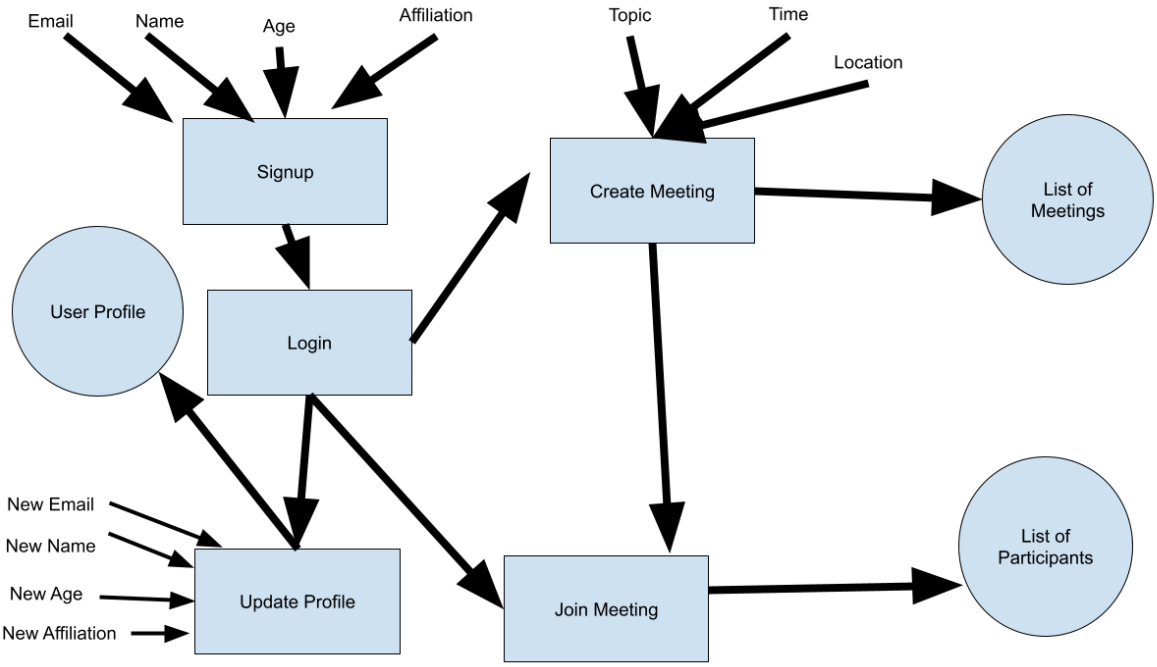
2. Preface

The version of this document is 1.1. We have updated this document from the previous version as we made adjustments to our detailed design section of the document. We added another class object to deal with a different functionality within the program. Furthermore, we added the finalized architectural design (Pipe and FIlter) to the document. Finally, upon finalizing the detailed design class diagram we made some substantial changes to the data flow found in some of the sequence diagrams at the bottom of the document.

3. Introduction

The purpose of the system we have chosen to help us have a contextual understanding of how we are to begin programming the application the client desires. We have separated our main functionalities into separated objects, that for the most part, all communicate with one another to make the application's features work smoothly. Finally, this system allows us to stay clean and efficient in the implementation of the application giving an optimal, fast, and pleasant user experience.

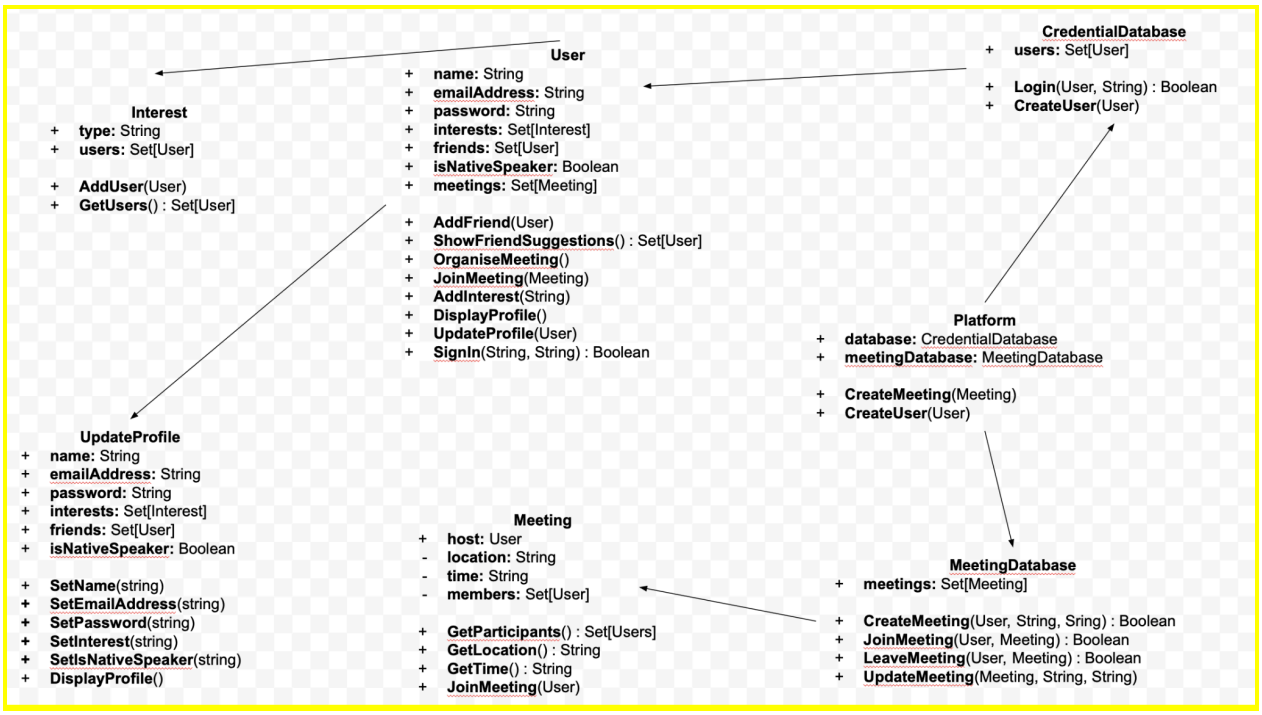
4. Architectural Design (Pipe and Filter)



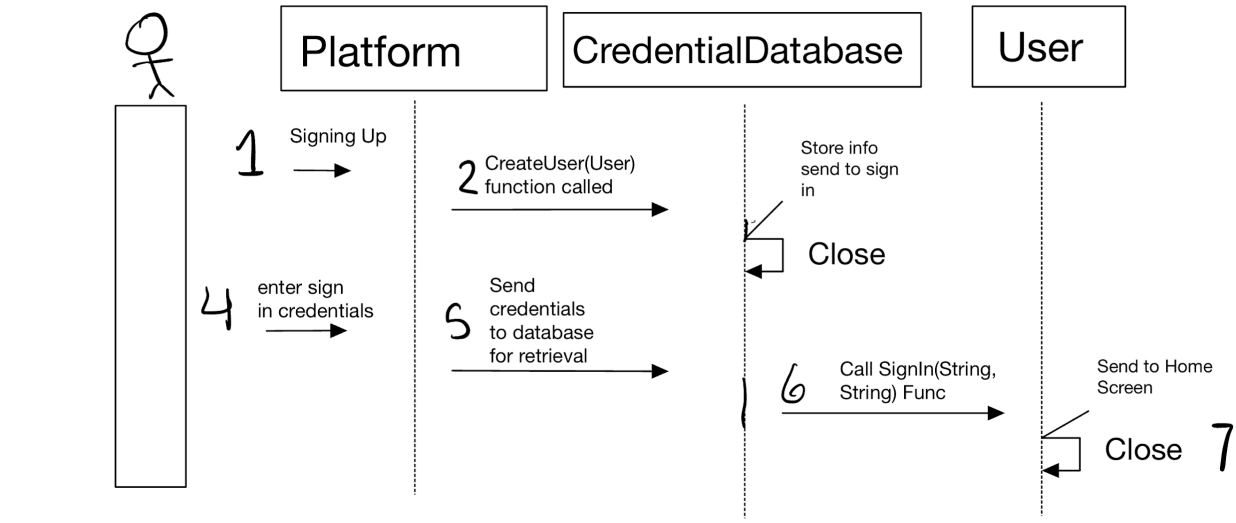
Components: Signup, Login, Create Meeting, Join Meeting

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.

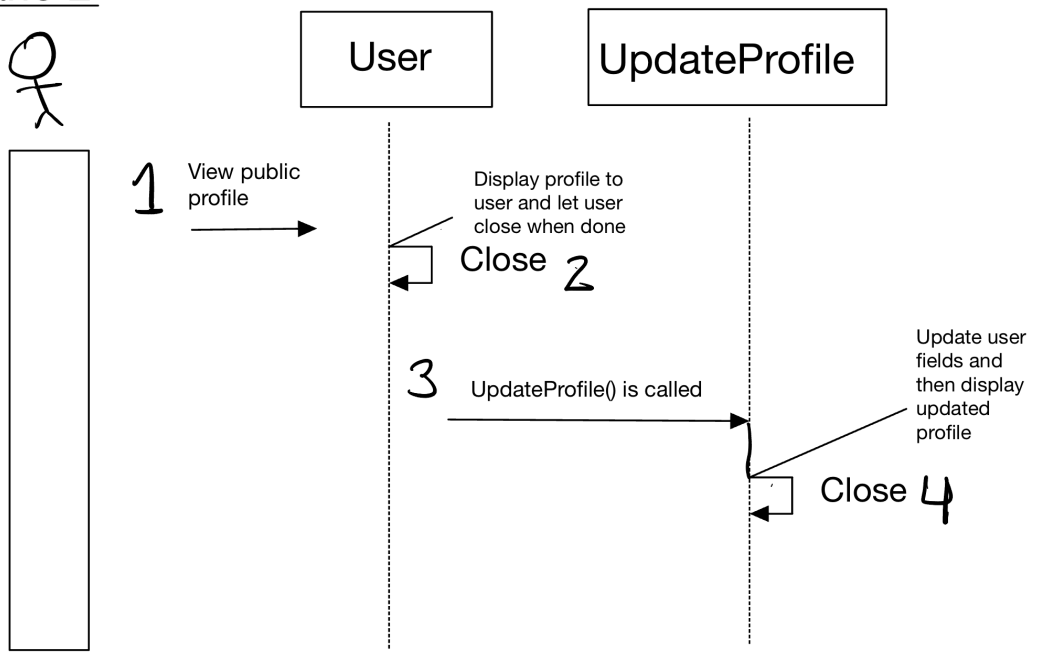
5. Detailed Design


Interest
+ type: String
+ users: Set[User]
+ AddUser(User)
+ GetUsers() : Set[User]
User
+ name: String
+ emailAddress: String
+ password: String
+ interests: Set[Interest]
+ friends: Set[User]
+ isNativeSpeaker: Boolean
+ meetings: Set[Meeting]
+ AddFriend(User)
+ ShowFriendSuggestions() : Set[User]
+ OrganiseMeeting()
+ JoinMeeting(Meeting)
+ AddInterest(String)
+ DisplayProfile()
+ UpdateProfile(User)
+ SignIn(String, String) : Boolean
CredentialDatabase
+ users: Set[User]
+ Login(User, String) : Boolean
+ CreateUser(User)
Platform
+ database: CredentialDatabase
+ meetingDatabase: MeetingDatabase
+ CreateMeeting(Meeting)
+ CreateUser(User)
UpdateProfile
+ name: String
+ emailAddress: String
+ password: String
+ interests: Set[Interest]
+ friends: Set[User]
+ isNativeSpeaker: Boolean
+ SetName(string)
+ SetEmailAddress(string)
+ SetPassword(string)
+ SetInterest(string)
+ SetIsNativeSpeaker(string)
+ DisplayProfile()
Meeting
+ host: User
- location: String
- time: String
- members: Set[User]
+ GetParticipants() : Set[Users]
+ GetLocation() : String
+ GetTime() : String
+ JoinMeeting(User)
MeetingDatabase
+ meetings: Set[Meeting]
+ CreateMeeting(User, String, Sring) : Boolean
+ JoinMeeting(User, Meeting) : Boolean
+ LeaveMeeting(User, Meeting) : Boolean
+ UpdateMeeting(Meeting, String, String)

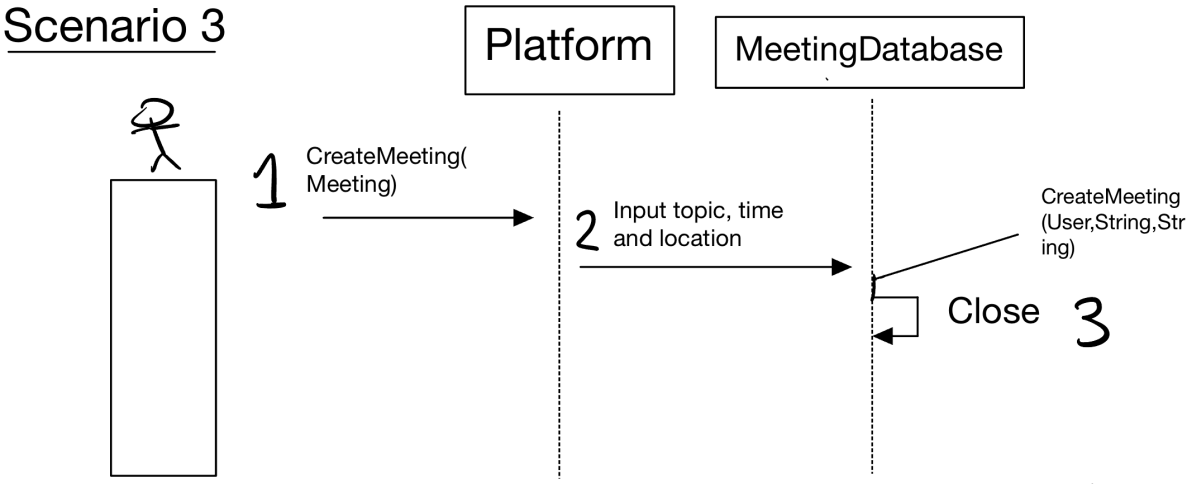
Scenario 1


Platform
CredentialDatabase
User
1
Signing Up
2
CreateUser(User)
function called
Store info
send to sign
in
Close
4
enter sign
in credentials
5
Send
credentials
to database
for retrieval
6
Call SignIn(String,
String) Func
Send to Home
Screen
Close
7

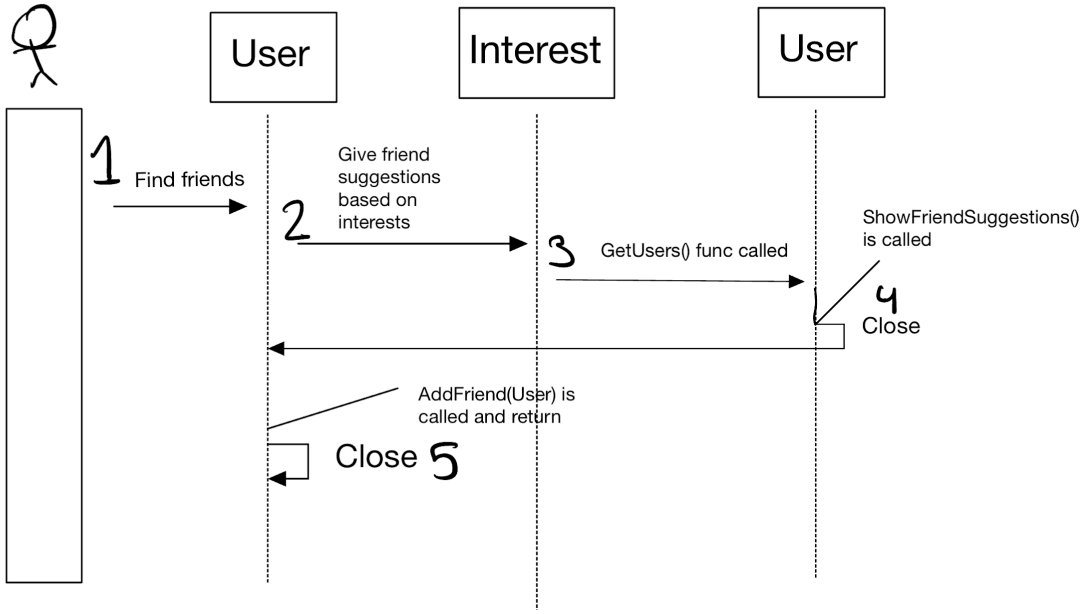
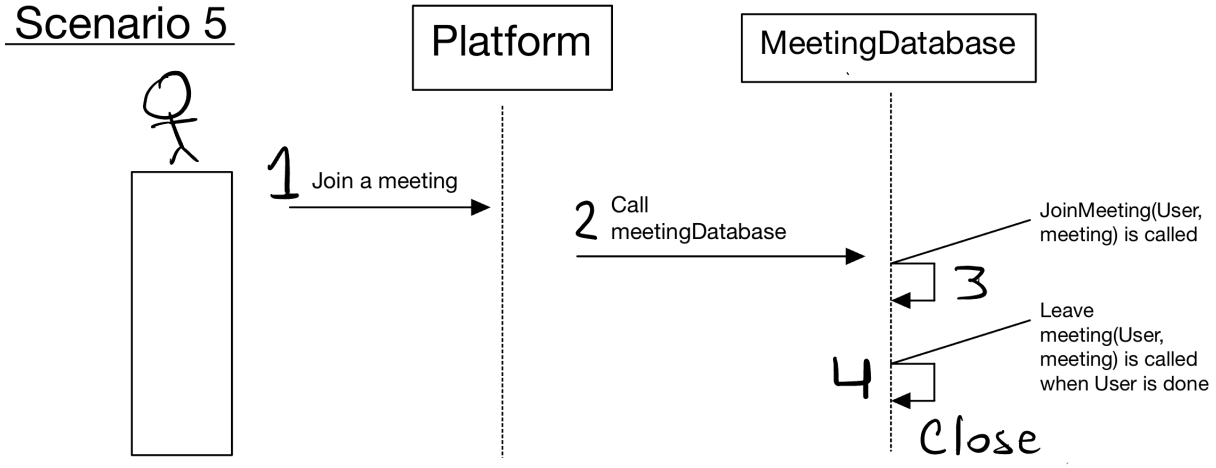
Scenario 2



Scenario 3



Scenario 4



Scenario 5