Users:

Primary Users:

- Silent Generation (70-100):
    - Primary users interested in meeting people through online dating within the age range of 70-100. They will be using the app directly to match with potential dates.
- Boomer (45-70):
    - Primary users interested in dating and meeting people through OkBoomer in the age range of 45-70. They will be using the app directly to match with potential dates.
- Babies (under 45)
    - Primary users looking to meet and match with people older than them through OkBoomer. They will be using the app directly to match with potential dates.

  We have decided to split up the primary users into these categories as their technological experience and dating preferences can vary drastically within the age range of 'elderly'. Similarly, we don't want to restrict users from signing up (provided they are legal), therefore, we need a category for those younger than 45 looking to date older people.

Secondary Users:

- Children/Grandchildren of Primary Users
    - Children/Grandchildren of primary users may need to help and motivate their parent/grandparent in the use of the app. Therefore, they can classify as a secondary user. They will be using the app, but not for themselves, rather they will use the app in order to help a primary user.
- Developer
    - The developers need to use the app, not to date and meet people, but in order to test the functionality of the app. Therefore they classify as a secondary user.

Requirement gathering technique and rationale:

We have decided to utilize a Questionnaire for requirement gathering. This is because we wanted to question users in the age ranges specified above. Since we would not be able to interview all of them in the time-frame we had and some were miles away, we decided to use a questionnaire. That way users could fill it out in their own time, and we did not have to travel in order to interview them. In addition, questionnaires let us analyze our data more effectively than an unstructured and even structured interview. Unfortunately, as this is a Questionnaire, the responses are somewhat up to interpretation and the nuances of responses are lost.

Users' Feedback:

Functional Requirements

Based on user responses, OkBoomer should filter potential matches by intentions (in terms of relationships), age, location, and interests. However, OkBoomer should not provide potential first date ideas or filters based on 'importance of family'. We should also provide a method of

ensuring security and protecting against catfishing as that is the main concern in online dating for both primary and secondary users. In terms of messaging, SMS texting was the main method selected by primary users, with the second method being in-app. Based on this, OkBoomer should ensure user validity and then exchange numbers. For user profiles, we need to have at least a biography and a list of interests.

Data/Resource Requirements

We will need to store a variety of information. For each user we will need to store their profile information (pictures, interests, biography, location, phone-number) as well as who they have matched with. If we implement in-app messaging, we will also have to store the messages between users.

We will be storing the data in a MySQL database (hosted on a server). Each user instance will have the attributes userID, age, location, phoneNumber, name, biography. We will save images with the attributes userID and order. We will then store matches in a table with the attributes MatchID, User1ID, User2ID (there will be two tuples per match). If we implement in-app messaging, we will then store the conversations in a table with attributes MatchID, and conversation.

Accuracy is key, and the size of the data will increase exponentially depending on the number of users. By using a MySQL database (hosted on a server), we will have high consistency and high availability, but the scalability suffers as size increases. One way to improve the use of size is to use a graph database for conversations to improve Partitioning, although the consistency of the data suffers.

Environment requirements

The app will be expected to operate at maximum functionality with WIFI or data and at minimum functionality without wifi or data.

The app will have to be usable anywhere a phone can be used with the functionality expected depending on wifi/data status.

In terms of use, any data sharing/messaging can happen asynchronously. The app will run on Android platform and will utilize MySQL (hosted on a server) and Java.

For this reason, the app should only be dependent on the environment via the Android mobile phone itself. Physical circumstances, such as light and noise, should not have an effect on the usability of the app. However, extreme conditions, such as extreme heat, cold, or light, that affect the usability of the mobile phone will be inherited to the app.

Usability requirements

The app should be able to deal with errors appropriately and efficiently. The system should be learnable and memorable for elderly users. The app should be efficient, allowing matching with the minimal number of clicks. The app should have larger text and buttons, to ensure that the older demographic can properly use the app. Above all, users should feel safe when using the app.

Two scenarios:

Scenario 1:

A registered user of OkBoomer, Val, opens the application. The user logs in and starts going through potential matches. Val presses the “OkBomb” button to say no to a potential and a heart button to say yes. While swiping, Val finds a potential she is interested in. She clicks the info button (an i) to read their bio. After that she swipes through their photos. She decides she is not interested and clicks the bomb button. Val continues until she finds a potential match she is interested in. She clicks yes and it's a match! She then sends a selfie to the match and receives one in turn. Once she receives the selfie, she can see that the person is who they say they are. She approves the selfie and they can now exchange numbers/messages.

Scenario 2:

Lee downloads OkBoomer for the first time and opens the app. They are brought to a create account page and start to create their profile. Lee adds their information as they are guided through the create account steps. They ensure that they have their required bio and interests, as well as the recommended photos. They also set their filters on age, location, and interests. Now that their account is created, they start to go through potential matches by reading their bios, looking at their interests and pictures. After a few profiles Lee finds someone they are interested in. They click the heart button. After a few minutes, Lee gets a notification about a new match. It was the person they liked! Lee sends the requisite selfie, but the selfie they receive is not of the person they thought it was. Lee is not in communication with the person they thought they were. Lee disapproves of the selfie. Lee and the person are unmatched and a notification is sent to the management team for them to look into the reported user.