

Team 7 Final Project Report

Title: Bubl

Team Members:

Victorina Joy G. Santos, Nina Boonyasiti, Kerr Salabao, Thang Le, Linh Nguyen

Synopsis: Interactive mobile dating app that provides a gameshow-like experience, allowing users to find more meaningful matches through personal compatibility.

Project Description: A large problem surrounding modern dating apps is the issue of excessive emphasis on physical aesthetics, leading to less meaningful matches and wasted time for those seeking a committed relationship.

By allowing users to ask initial matches in their “bubl” a variety of questions, emotional compatibility can be found when a user “likes” an anonymous response from another user. Users may leave or swap their incompatible matches to get a new set of anonymous matches in their dating pool or “Bubl”. By allowing users to ask their anonymous matches questions about topics most important to them, it can cut the wasted time out of dating while removing visual bias. Additionally, users will have a set of initial arbitrary recommendations according to the data shared upon creating their account (this will include dating preferences, sexuality, baseline information, etc.).

After “liking” a response to a posted question, a candidate has the opportunity to be part of a user’s Bubl. By becoming one of the ‘top matches’ users will be able to reveal each other’s profile pictures and unlock a private chat. To summarize our expected results, by first building emotional compatibility, we hope to ensure more meaningful and long-lasting matches.

Project Milestones:

Senior Project Status Reporting and Milestone			
Date	Milestone	Status	Changed
Fall 2021			
September 24, 2021	Project requirements defined	Completed	
October 1, 2021	Initial Project Designed	Completed	*
October 8, 2021	Rough Draft for Figma design and Final Report	Completed	*
October 15, 2021	Figma Design completed	Completed	*
October 22, 2021	Planned Project Video Presentation	Completed	
October 29, 2021	Video script rough draft + Final Report completed	Completed	*
November 5, 2021	User Sign Up implementation + Video recorded	Completed	
November 12, 2021	Video completed	Completed	*
November 19, 2021	Permanent UI Design to Argon + Import dependencies and libraries	Completed	*
November 26, 2021	Integrating Argon to current screens	Completed	*
December 3, 2021	Database installed	Completed	
December 10, 2021	<u>User Authentication completed</u>	Completed	
Spring 2022			
January 21, 2022	Set up weekly team meetings, GTA meetings & update Github issues	Completed	*
January 28, 2022	User profile upload completed	Completed	
February 4, 2022	Initial Dashboard design completed	Planned	
February 11, 2022	Mobile front-end styled	Planned	
February 18, 2022	Matching Algorithm completed (1/2)	Planned	
February 25, 2022	Matching Algorithm completed (2/2)	Planned	
March 4, 2022	Question-making function implemented(1/2)	Planned	
3/11/2022	Question-making function implemented(2/2)	Planned	
March 18, 2022	Chatting function implemented (1/2)	Planned	
March 25, 2022	Chatting function implemented (2/2)	Planned	
April 1, 2022	Basic app features completed	Planned	
April 8, 2022	Front-end design improved	Planned	
April 15, 2021	Improved Front-end design implemented	Planned	
April 18, 2022	Project deployed	Planned	
April 22, 2022	Project video presentation recorded	Planned	
4/29/2022	Project video presentation completed	Planned	

Gantt Chart:

TASK TITLE	TASK OWNER	FALL 2021											
		SEPTEMBER			OCTOBER			NOVEMBER			DECEMBER		
		9/24	10/1	10/8	10/15	10/22	10/29	11/5	11/12	11/19	11/26	12/3	12/10
Project requirements defined		40.6											
Help define requirements	Kerr	11.2											
	Nina	9											
	Thang	11.2											
	Linh	11.2											
	Joy	4											
Initial Project Designed			40										
	Kerr		8										
	Nina		8										
	Thang		8										
	Linh		8										
	Joy		8										
Rough Draft for Figma design and Final Report				40									
	Kerr			8									
	Nina			8									
	Thang			8									
	Linh			8									
	Joy			8									
Figma Design completed					40								
	Kerr				8								
	Nina				8								
	Thang				8								
	Linh				8								
	Joy				8								
Planned Project Video Presentation						40							
	Kerr					8							
	Nina					8							
	Thang					8							
	Linh					8							
	Joy					8							
Video Script + Final Report completed							40						
Video Rough draft + final report	Kerr						8						
Video Rough draft + final report	Nina						8						
Login/signup navigation + final report	Thang						8						
Login/signup navigation + final report	Linh						8						
Login/signup navigation + final report	Joy						8						
User Sign Up implementation + Video recorded								40					

Gantt Chart (Cont.):

TASK TITLE	TASK OWNER	FALL 2021											
		SEPTEMBER			OCTOBER			NOVEMBER			DECEMBER		
		9/24	10/1	10/8	10/15	10/22	10/29	11/5	11/12	11/19	11/26	12/3	12/10
User Sign Up implementation + Video recorded								40					
Video recording + revise final report	Kerr							8					
Video Recording	Nina							8					
Video Recording	Thang							8					
Video Recording + navigation bar	Linh							8					
Video recording user profile	Joy							8					
Initial front-end coding (2/2) completed									40				
	Kerr								8				
	Nina								8				
	Thang								8				
	Linh								8				
	Joy								8				
Initial back-end coding (1/2) completed										40			
	Kerr									8			
	Nina									8			
	Thang									8			
	Linh									8			
	Joy									8			
Initial back-end coding(2/2) completed											40		
	Kerr										8		
	Nina										8		
	Thang										8		
	Linh										8		
	Joy										8		
Database installed												40	
	Kerr											8	
	Nina											8	
	Thang											8	
	Linh											8	
	Joy											8	
Mobile front-end styled													40
	Kerr												8
	Nina												8
	Thang												8
	Linh												8
	Joy												8
Total Hours:													440

Project Budget:

Software/Hardware needed	Cost	Vendor	When they are needed?
Repo manager	\$0	Github	10/1/21

Live development Environment	\$0	ExpoGo	10/23/21
Database management	\$0	Firestore	11/20/21
Front-end Framework	\$0	React-Native	10/18/21
Back-end Framework	\$0	Firebase	11/20/21

Team 7 Final Project Proposal

Team Number: 7

Team Members: Victorina Joy G. Santos, Nina Boonyasiti, Kerr Salabao, Thang Le, Linh Nguyen

Preliminary Project Design

I. How the software works:

1. Project workflow (Figure 1):

To begin with Bubl, users can register/sign-up using their personal emails or Google account. Upon their first registration, users will be asked for personal information such as name, age, gender, sexual orientation, location, preferred age, etc. Once submitted, their personal information can be updated in the user profile section.

After the authentication process, users will be led to the dashboard. Initially, a user's Bubl is empty. Users will choose either choose questions already generated in the app or create a customized question. After posting a question, users will be able to see anonymous answers from other candidates in their local area. It is entirely of the user's choice to choose up to five candidates they would like to be in their Bubl. After scrolling through the answers to a user's question, a user can swipe right or "like" a response. The candidate is then added to the Bubl. From here, users and the candidates in the Bubl have unlocked profile pictures and the capability to chat privately. The user can then post more questions to their anonymous candidates and if another response satisfies them, they can

have the option of either adding to their Bubl or swapping a current unsuitable candidate with a new candidate into the Bubl. To remove a user or undesirable candidate (say, answers are unsatisfactory), users may swipe left on their answer, removing them from the candidacy pool. If a candidate is removed from the Bubl, that user becomes anonymous – however any chat logs with that user are kept.

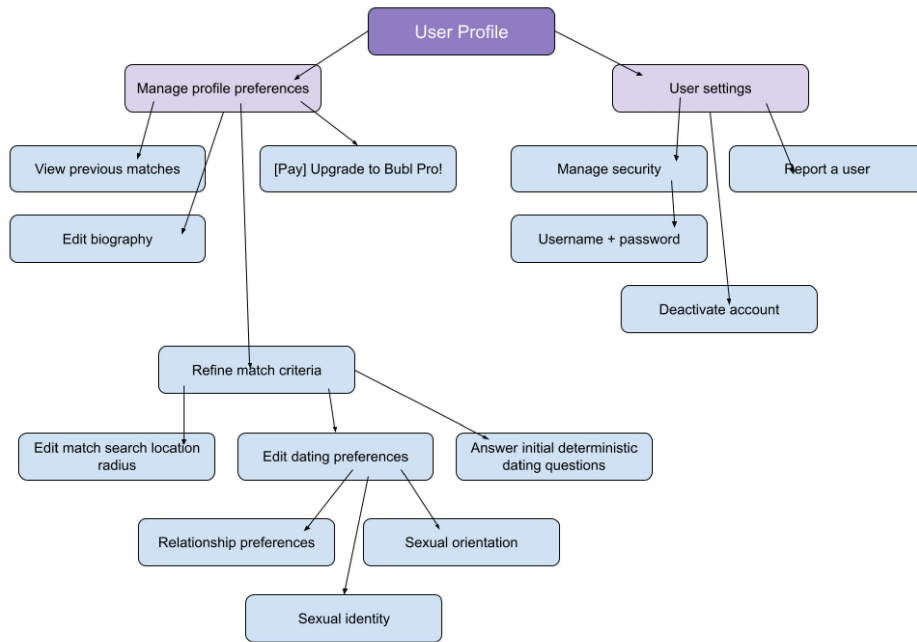


Figure 1: Project Workflow

The goal of Bubl is to create a game show-like experience with in-depth emotional compatibility that no other dating applications have.

2. Backend:

The backend portion of the app is implemented to handle user events and maintain the state and data of the application. This means the backend should contain two components: event handler and database system. The backend system handles a specific event and performs the related actions, often involved with querying the database system. For example, users can upload their personal information like sexual orientation, location, age, profile picture, etc. The backend system can handle the “upload” event of the user and store information of that user to the database. The user can update their information later and the database will maintain this information in real-time (Figure 2). Data will be in real-time retrieval in order to guarantee that the data a user has is factored into the quality of matches they receive.

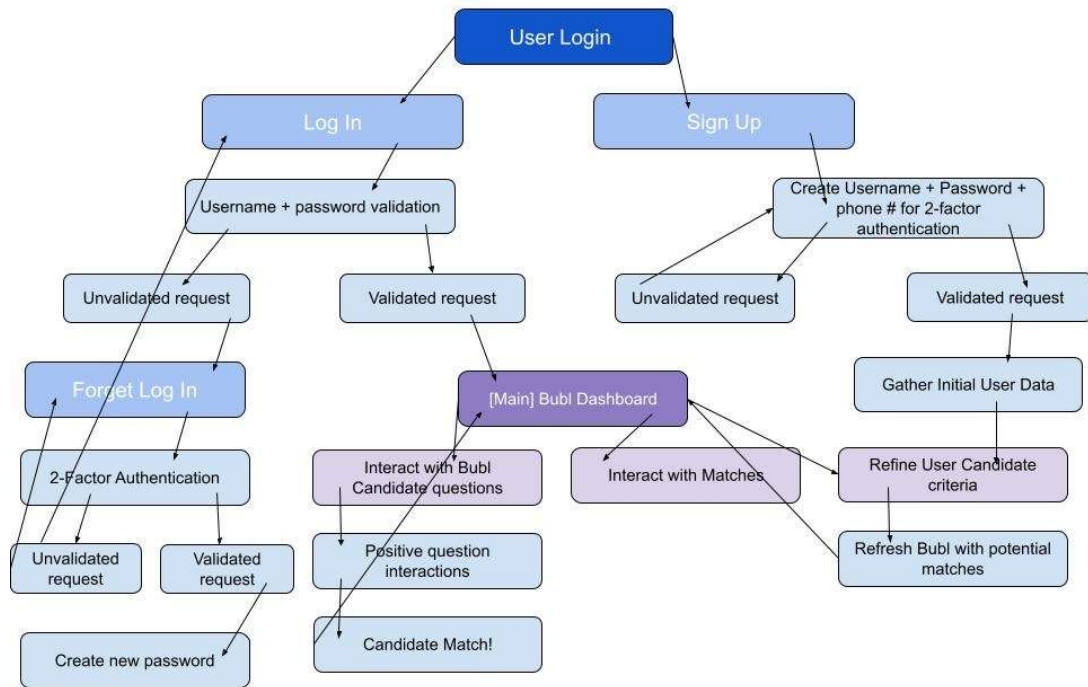


Figure 2: User information backend

Another example that can be listed is Matching Backend. When a user finds their good match and wants to start the conversation with the partner, we would want the backend to save the state of their conversations. (Figure 3)

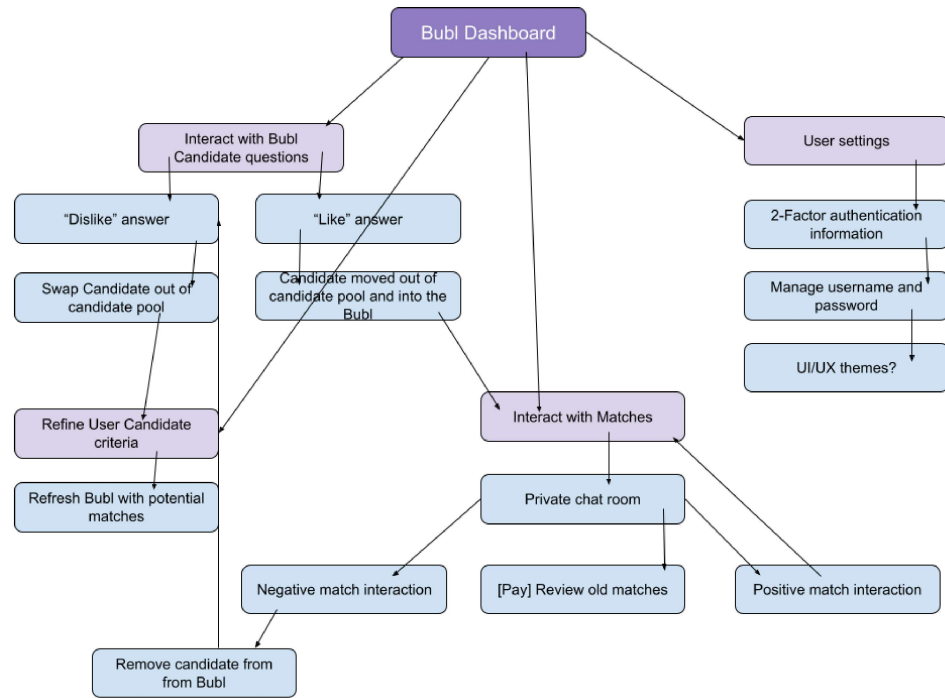


Figure 3: Matching backend

Even though there are many ways to implement a backend system, we will use Firebase, a Google-backend application that enables developers to develop IOS, Android, and web apps. Firebase provides tools such as authentication, Firestore real-time database, reporting and fixing app crashes. In addition, Firebase is recommended to build a real-time chat application, which is one of the main features of our project, by virtue of the Firestore real-time database. With the Firestore real-time database, real-time retrieval is possible for each time there is a change in the database, which keeps users up-to-date with their latest matches. Thus, we believe Firebase is the right platform to deploy Bubl in an efficient way in a short period of time, given our tight schedule.

3. Frontend:

We strive to create a solution that is flexible and accessible for both iOS and Android platforms. Thus, we shall use React Native for this project as it allows developers to write the project in JavaScript and renders it in native code that can be used for creating Android and IOS applications. Our goal is to create a simple and user-friendly, but elegant frontend.

To design Bubl, the Argon Design System will be used as a guide for the look of the app. However, the main framework that will be used for the frontend is UI Kitten. While Argon will be the design implemented, UI Kitten will provide the frontend framework as UI Kitten proves to be the better choice for the project's needs.

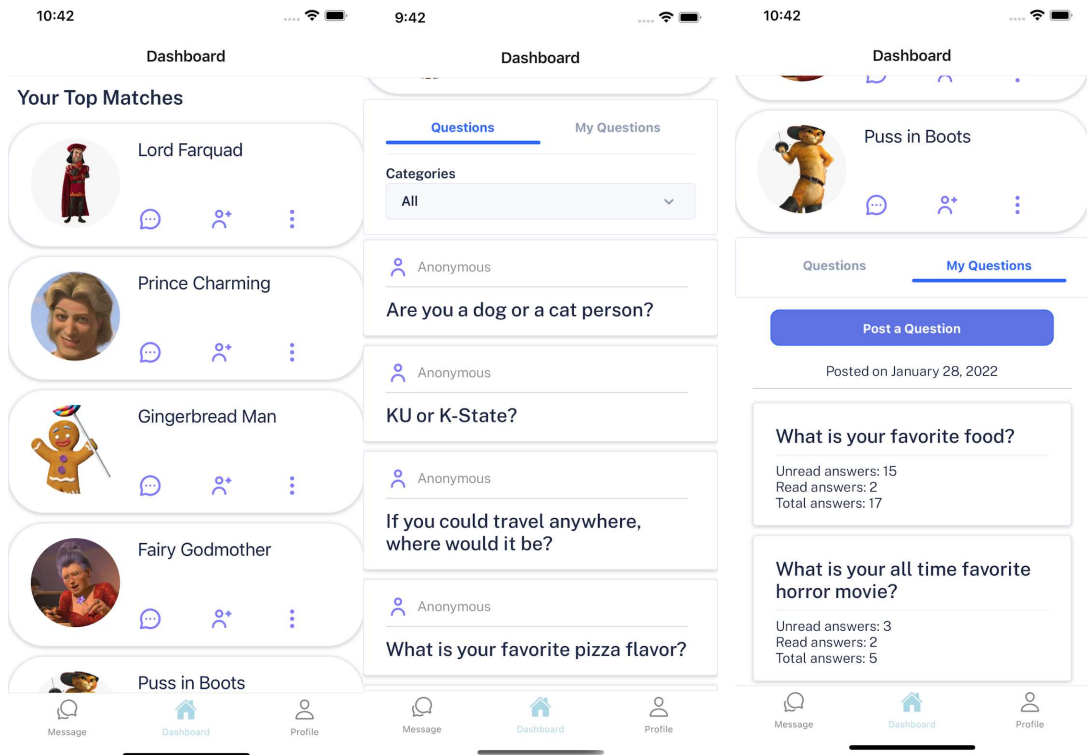
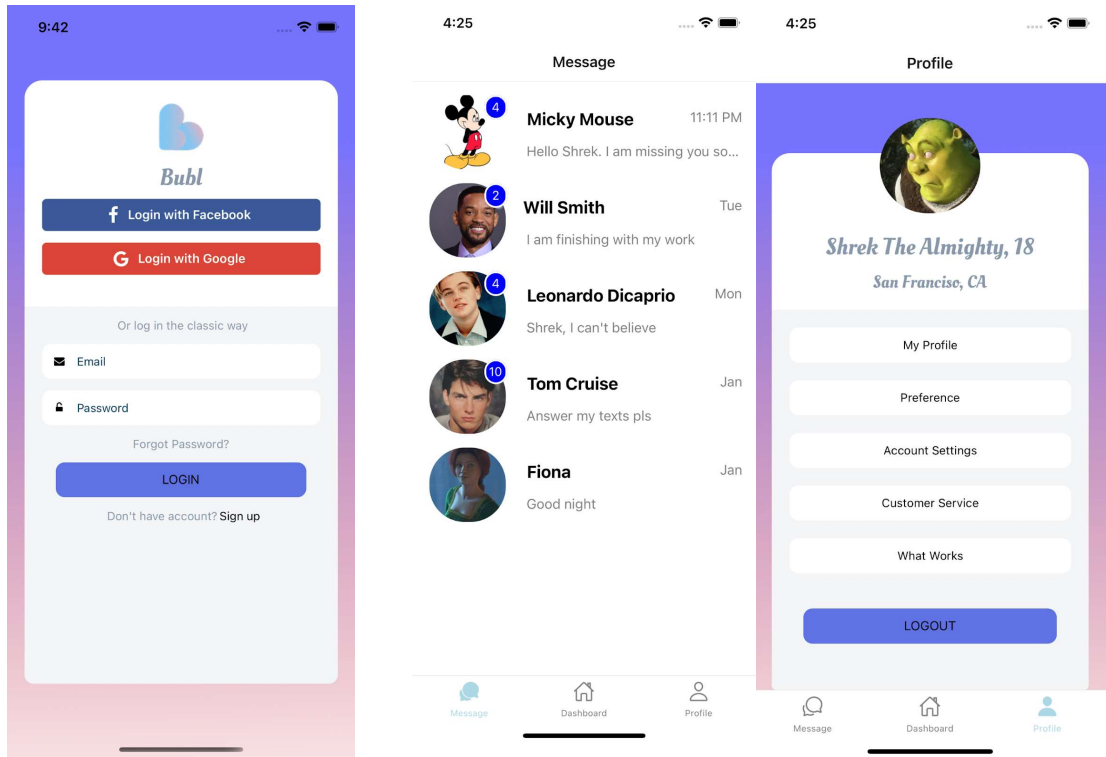


Figure 4: Argon Based Design

Another technology that we will be utilizing for development is ExpoGo. ExpoGo is an open-source platform that makes developing cross-platform iOS and Android mobile apps much easier. It provides a preset SDK (Software Development Kit) with commonly used APIs like basic view components, images, camera access, notifications, device info, and more. With ExpoGo, it enables us to see the changes real-time in our personal mobile devices or through different devices via an emulator whenever we update our source code.

4. Testing Methodology:

Functional Testing ensures that the project is functioning well and correctly with assigned requirements and specifications and in line with user expectations. This testing focuses more on the flow of the app, and it ensures that all the features are responsive and meet the required specifications. Functional Testing should make sure that the application installs and launches correctly, that users can sign-up and login, and that text boxes and buttons function properly.

Usability Testing checks the user experience with insights that include identification of bugs and make improvements for customer understanding. Usability Testing should include a set of test scripts and feedback questionnaires for each task and feature of the project for recommendations enhance customer service. This is a great opportunity to discover opportunities and learn about users.

Our application asks for the user's personal information. Therefore, Security Testing must guarantee confidentiality, authenticity, and integrity of the app. The main goal is to identify the threats in the system and measure its potential vulnerabilities, so that the threats can be resolved to prevent the system from crashing.

II. Design Constraints:

1. Technical constraints:

- Programming languages: JavaScript, React Native and Firebase
Operating system or platforms supported: iOS and Android. Mobile Applications are convenient, quick and accessible. A private instance of an app on a user's phone offers privacy and freedom – this becomes a source of entertainment that can easily be shared across a variety of devices.
- Use of specific library or framework: React Native, Firebase. React Native is a framework building front-end for mobile applications along with iOS and Android. Firebase is a platform that offers cloud-based services for creating the back end for our mobile app that lets us store and sync data

between users in real time. React Native is supported through binding to the database.

2. Business Constraints:

- Schedule: The fact that we all have other classes and some of us also have part-time jobs. It's totally up to us to design a system that can be built within the required schedule. In order to do that, we have to consider our milestones and set the deadlines for each requirement that we need to do.
- Team composition and make-up: If someone can't show up in the weekly meeting, we want to keep them up to date with whatever team members are working on. Members must update it in the meeting or let everyone know in the group chat.
- Design Schedule: project planning, project control
- Development Schedule: designing detailing, compliance tests
- Budget: Design Cost, React Native training
- Software licensing restrictions or requirements:

3. Aesthetic Constraints:

- User Needs: type of operation, instructions, and warnings
- Customer Appeal: shape, color, texture, form, delight features
- Fashion: culture, history, trends
- Future expectation: rate of change in technology

III. Ethical Issues

Ethical issues for Bubl vary from interpersonal interactions to logistical operations. Issues that could arise from user-to-user interaction include discrimination and hate speech. Although these issues are fairly straightforward and easily identifiable, complex issues such as Catfishing (the act of pretending to be another person) and Ghosting (the act of feigning interest and then ignoring the person of interest) require more specific and thoughtful solutions.

1. Freedom of Speech & Hate Speech

- a. A potential solution to user-to-user issues include moderating conversations to strike a user's account when discussing topics with high-risk activity, such as activities related to harming others/one's self. Although this is a common method used on various social media platforms, a major issue that comes up with this is user confidentiality and the obstruction of free speech. Ultimately, Bubl strives to provide a safe and inclusive experience for our users — so a compromise could be made to moderate interactions and let the user know of this fact while signing up for the app (privacy policy).

- b. As for the ethical issue of freedom of speech and what can/cannot be moderated, Bubl has decided to moderate conversations and interactions and assess the level of risk these interactions have. Risk assessment is subjective, but a conversation or interaction will be marked as inappropriate if discussions or behaviors around harming the physical or emotional well being of one or more persons. Physical and emotional well being may be defined as but not limited to the integrity of another person's character, financial wellness, mental wellness and reputation.

2. Catfishing & Ghosting

- a. Another solution for more complex issues like Catfishing include user id verification. While our user base is limited to the University of Kansas's Engineering student population, an alternative to requiring personal identification such as Jumio could be to require a student's KU ID number. We can then use these given KU ID numbers to verify that this student is who they claim to be. When an initial match is made, there is some form of accountability, because users may be able to report other users, helping our team to find the root of the issue and take the necessary steps to ban a user or report them to authorities.
- b. Issues surrounding complex interpersonal relationship behaviors such as Ghosting require the identification of where this behavior stems from. Feigning interest of another may be rooted in the need for one to be desired or for any other reason -- Bubl's conclusion is that the feigning interest of others comes from the poor quality of initial matches. If users are matched with other incompatible users, an initial feigning of interest is created and eventually leads to mutual silence or ghosting. With Bubl's algorithm to match users based on similar answers regarding intent (dating preferences), sexuality, ethical values, goals and personality, Bubl seeks to create matches that stimulate emotional and mental compatibility and therefore more meaningful matches.

3. Discrimination

- a. Discrimination may be defined as the act of unjust prejudice of a person or persons based on but not limited to the grounds of race, age, and sex. Discrimination is a form of hurtful behavior that is present in verbal and non-verbal communication. Online behaviors will be more difficult to moderate, but a risk assessment will be performed to recognize microaggressions and discriminatory actions and behaviors. Bubl is currently handling complex interpersonal issues among users on a case by case basis, but users will be made aware of this form of moderation in the privacy policy.

IV. Intellectual Property Issues

There are cases of intellectual property issues to be addressed. The cases can be split between open source and non-open source. While all of the software that is being used is free, there are caveats to the non-open source software that will be used.

1. Open Source

- a. Starting with the front end with **React**. React is an open source Javascript framework. Since it is open source we are free to use it for our project with no issues regarding intellectual property.
- b. The other open source software that will be used is **UI Kitten** for the UI/UX. UI kitten is under the CC BY 4.0 which states

*“Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material
for any purpose, even commercially.”*

There is the responsibility to give credit to the creators, but aside from giving credit there are no other caveats to the intellectual property that needs to be addressed.

2. Non-Open Source

- a. The first non-open source software that will be mentioned is **ExpoGo** which will be used for hosting the mobile platform for both Android and Apple. ExpoGo has this excerpt regarding their policy on intellectual property:

“We claim no intellectual property rights over the material you provide to the Service. Your profile and materials uploaded remain yours. ”

This means that all of the work that is uploaded on the app will have their creators reserve all intellectual property rights.

- b. The next software is **Firestore** which will be used for backend, user authentication, and database. Their intellectual property rights is as follows:

“ As between the parties, Customer owns all Intellectual Property Rights in Customer Data and Customer Applications, and Google

owns all Intellectual Property Rights in the Services and Software.”

meaning that the customer retains all rights to the work that they created through the tools and services that Firebase provides. However the firebase software and by extension the services that they provide is strictly under Google’s Intellectual property.

- c. The last software that will be used that is not open source is **Figma**. Figma will be used for designing what the app will look like. Figma follows a similar rule as ExpoGo and Firebase does, with this excerpt claiming that
“Figma, Inc. (“Figma”) respects the intellectual property rights of others and expects its users to do the same.”

meaning that just like ExpoGo and Firebase, the creators retain the rights to their personal work, but not to the software that was used to create those work themselves. However Figma does have an added caveat which states

“It is Figma’s policy, in appropriate circumstances and at its discretion, to disable and/or terminate the accounts of users who repeatedly infringe or are repeatedly charged with infringing the copyrights or other intellectual property rights of others.”

warning users that repeated copyright and intellectual property right violations could lead to services being denied to the user.

V. Change Log

1. User Data

- a. During the time we were in the process of generating ideas for our senior design project, we initially envisioned a dating app that utilizes user data that is sourced from their personal social media accounts such as Instagram, Facebook, Twitter, and Youtube. The data contains social media content that they have reacted to with ‘likes’ or anything of that equivalence. Users will upload their data into the app, which would then search matches for them according to their shared interest.
- b. By following this route and using social media platforms as a third-party source of user data, we come to a problem for accessibility. Data collection from other social media platforms is difficult or impossible due to the privacy policies and rules that are imposed by each of the social media platforms. To solve this problem, our team decided to let users create their own matchmaking algorithm through a gameshow-like process. The process involves users asking questions and answering questions from one another, and choosing the best answers out of the rest.

This process establishes a point-system which lets users find their best match.

2. Identity Verification

- a. One of the main problems that arise in some online dating platforms is catfishing. Catfishing is a term for describing an instance where an individual is lured into a relationship by someone who is using a false identity. To prevent this problem from arising, our team decided to use Jumio's identity verification solutions as our third-party solution to the problem. However, upon reaching out to Jumio for more information about the solutions they offer as well as their costs, our project's budget fell short. The cost of using their services has a base price of around \$3,000 with an additional \$3 for every user that uses identity verification.
- b. As an alternative, our team decided to collect KU student IDs from users. Even though verification is not possible due to the lack of accessibility to the KU student ID database, we will still collect KU IDs to hold users accountable if issues arise.

3. Backend and Database

- a. The initial technologies involved in our project included the usage of MongoDB as our database. However, after discussions regarding security concerns and authentication, we have decided to use Google's cloud services, which is called Firebase, in order to generate and handle user accounts, logins, and signups. Firebase has libraries dedicated to authentication, as well as a database, which is called Firestore. By using Firebase, it saves us time and resources on trying to implement secured authentication from scratch. Additionally, we decided to extend our dependence on Firebase to not only use it for authentication but also for database and backend procedures, so that our codebase is more consistent and hopefully reduces the amount of new technologies we have to learn.

4. Frontend design

- a. The design for the frontend has been changed from the original Figma design. With the exception of the dashboard most changes were primarily the aesthetics of the app. The aesthetics were modeled off the Argon design system.
- b. From the original design, the intention was to have the button layout inside a circle to emulate a bubble. However, to fit the theme of Argon adjustments had to be made and the circle button layout were tabled for later.

- c. In addition to the different layout for the dashboard, some major functionality changes were also made. The dashboard now contains questions the user can answer and responses to the user's own question.

5. Matching Algorithm

- a. Originally the matching algorithm had a point system in which each favorable answer to a user's question earned points. Earning enough points from the user would then get placed into the other user's "Bubl" unlocking chatting functionality. The new matching algorithm gives the users more flexibility in who they choose to be in their "Bubl" by letting them choose from a list of users that have answered their questions. The users can also freely remove users from their "Bubl" in order to add new users into it and unlock the ability to chat with them.