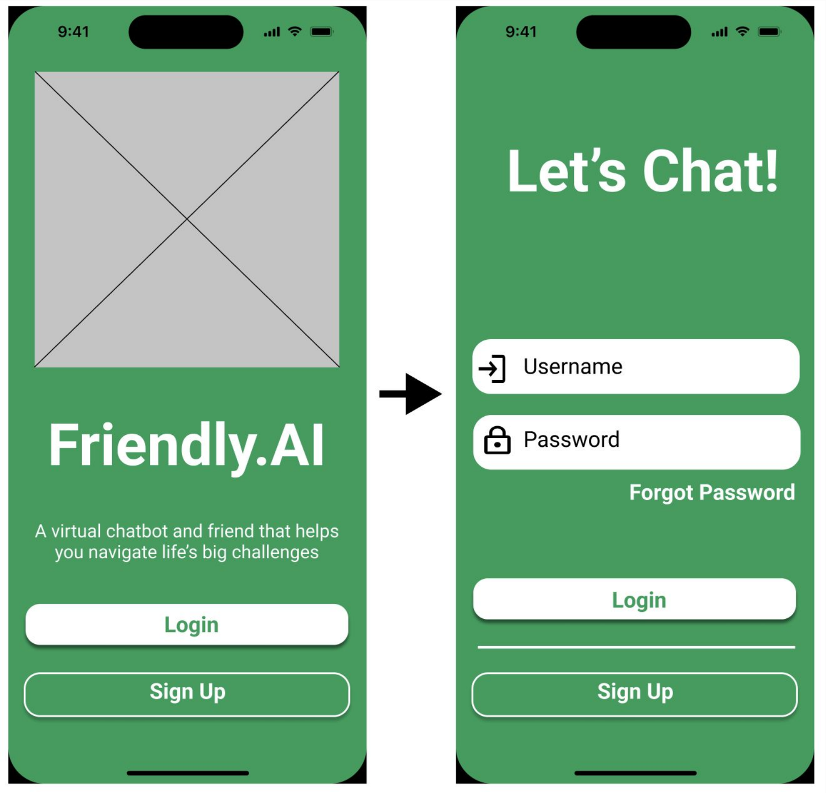
Friendly.AI
Solution Prototype + Social Affordances

“A virtual chatbot and friend that helps you navigate life’s big challenges”

1. Low-Fidelity Prototype

Friendly.AI Wireframe

- We start with a nice **landing page** and a nice **login page**.



A virtual chatbot and friend that helps you navigate life's big challenges

Friendly.AI Storyboard (1/3)

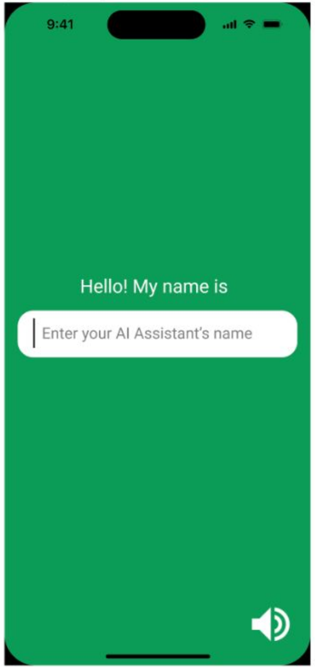
- We have five different steps with respect to the storyboard that showcases the user experience:

1. Prompting for the AI’s name through voice.
2. Prompting for the user’s name through voice.
3. Giving the user the ability to customize the AI chatbot.
4. Specific features such as **gender** and other features can be customized for our AI chatbot.
5. Once a specific feature is delineated, the user can the customize all the other features.

- By going through these five steps, we will have a conversational chatbot ready to go!

Friendly.AI Storyboard (2/3)

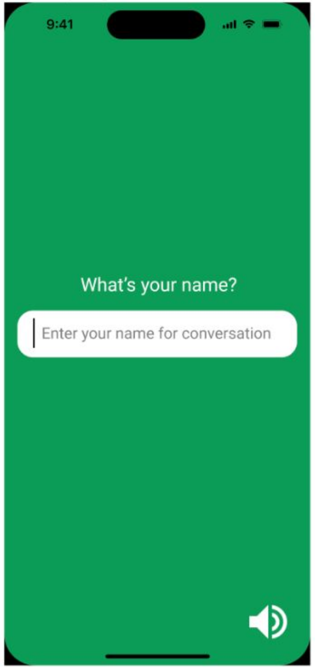
1) Prompting for the AI's name.



Hello! My name is

Enter your AI Assistant's name

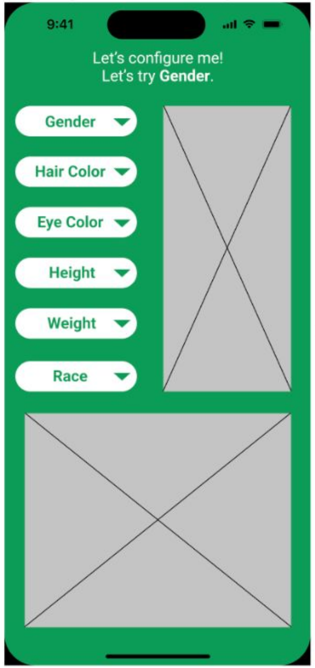
2) Prompting for the user's name.



What's your name?

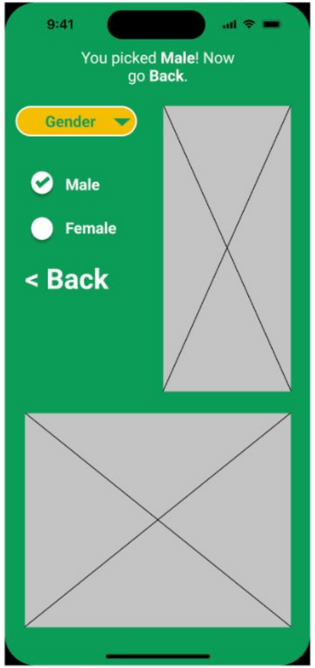
Enter your name for conversation

3) Giving the user the ability to cu...



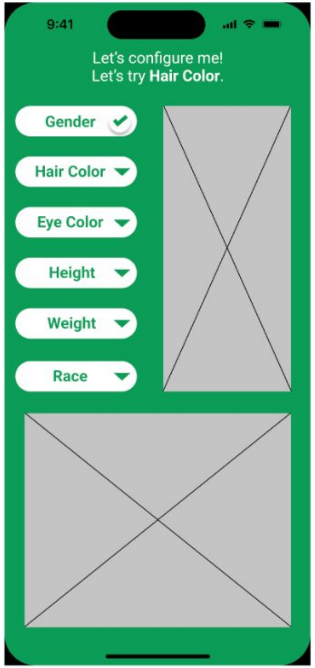
Let's configure me!
Let's try **Gender**.

4) Specific features such as gende...



You picked **Male**! Now
go **Back**.

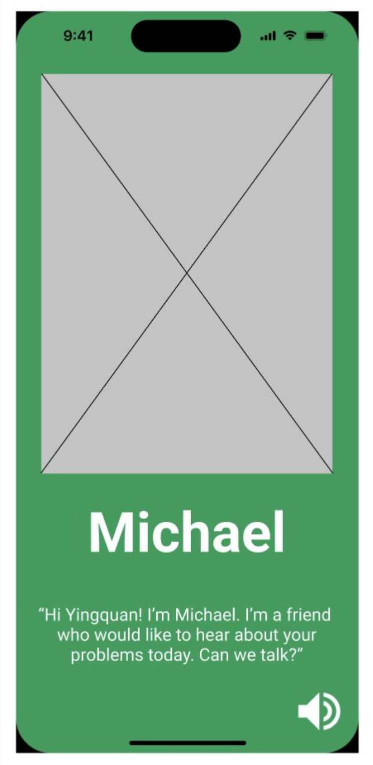
5) Once a specific feature is define...



Let's configure me!
Let's try **Hair Color**.

Friendly.AI Storyboard (3/3)

- Once the chatbot features have been customized, we can start talking to the chatbot through **voice**, with subtitles provided for the duration of the conversation!



Michael

“Hi Yingquan! I’m Michael. I’m a friend who would like to hear about your problems today. Can we talk?”

Wireframe & Storyboard Description

- The **Friendly.AI** application is intuitive and easy to use, with a sleek user interface that delivers an unforgettable customer experience.
  - **AI customization that's personalized**: The user is able to customize an AI chatbot and friend that appeals to him or her. The personalization will enable the user to trust the AI and feel like it's a real friend that the individual can share his or her problems with.
  - **State-of-the-art AI technology**: Leveraging *large language models* (*LLMs*) and OpenAI APIs for GPT4, the Friendly.AI chatbot will respond to users through voice over such that the conversation feels natural. The chatbot will give the user the feeling that they are being heard, listened to, and cared for.
  - **Data Security**: Users who store their personal information (users, passwords, demographic information, etc.) with **Friendly.AI** know that their data will be encrypted in-transit and at rest, thus providing robust data security.

- **Use Case #1**: A college freshmen guy is having problems fitting in socially at his new school. He's feeling depressed and alone. He uses **Friendly.AI** as a chat buddy and his AI friend encourages him to find friends that he can relate to in real life, giving him confidence. The guy has a whole new outlook about college.
- **Use Case #2**: A young man in his twenties moves to a new city where he doesn't know anyone. He is encouraged by **Friendly.AI** when he's alone to stay positive and keep going. He takes action and starts volunteering in the community, and thus starts building a positive social circle.

2. Social and Technological Affordances

User Interaction

- A user in the U.S. with a mobile phone will download the **Friendly.AI** application and then be able to use the app. The *user interaction* flow would be described as follows:

    1. The user would need to create an account as a first time user, providing some *PII information*.
    2. The user would then be authenticated using **multifactor authentication**, either through a *phone number* or *email*.
    3. The user would then customize the avatar and friend such that he or she is comfortable chatting with the avatar.
    4. Then finally, the user is able to start speaking to the avatar through voice and start a real conversation with someone who will listen to them 24/7.

- Once the avatar is created, the user will not be able to modify the avatar again, unless they delete their account and recreate their account. This is so that the user can't change their mind and is locked in to thinking that the avatar created is their actual friend.

- The reason why the user interaction is so simple is because the **simplicity is what makes the app accessible**! By limiting the features rolled out in the MVP, we will solicit feedback from early adopters on what they like or don't like about the app. We will also seek feedback from people who suffer from disabilities and would want them to be included in the **Friendly.AI** experience!

Social Implications

- **Friendly.AI** is ultimately an application with a social good. The tagline of the application is:
  - *"A virtual chatbot and friend that helps you navigate life's big challenges"*

- The hope is that through using **Friendly.AI**, millions of people will suffer from loneliness less and have a positive support system, even if the system is an AI.

- The **pros** include the following:
  1. The solution is very scalable and almost everyone has a mobile phone these days.
  2. There is a big market here in the U.S. for the product, should it be build in a way with a seamless *user experience*. It could potentially have viral/explosive growth.
  3. It would be easy to monetize eventually through ads, subscriptions, etc. after **critical mass** has been reached and people really enjoy the product.

- The **cons/ethical considerations** include the following:
  1. When people share very dangerous personal thoughts with the intention to harm either themselves or others, should the **Friendly.AI** team notify authorities?
  2. Will the app keep logs of people's conversations in the form of log data and store that? Is that a breach of privacy? What about the temptation to sell user data for profit? Must tread carefully!
  3. Will the LLMs that power the chatbots have hidden biases for certain demographics? Is the data that the LLM is trained on *fair* and *equitable* for all parties?

Technological Affordances

- In terms of *technological affordances*, the **Friendly.AI** app is very easy and intuitive to use for the consumer. A holistic *conceptual model* of the product was thought through carefully and sketched out on paper before any design work or app building commenced.

- The application is inherently *discoverable* and it's easy for the user to know what he or she should do once the app is installed from the app store. The designer (me) thought extremely carefully about the user experience, such that the app is DEVOID OF: **1)** *dark/deceptive patterns*, **2)** *false affordances*, or **3)** *hidden affordances*.

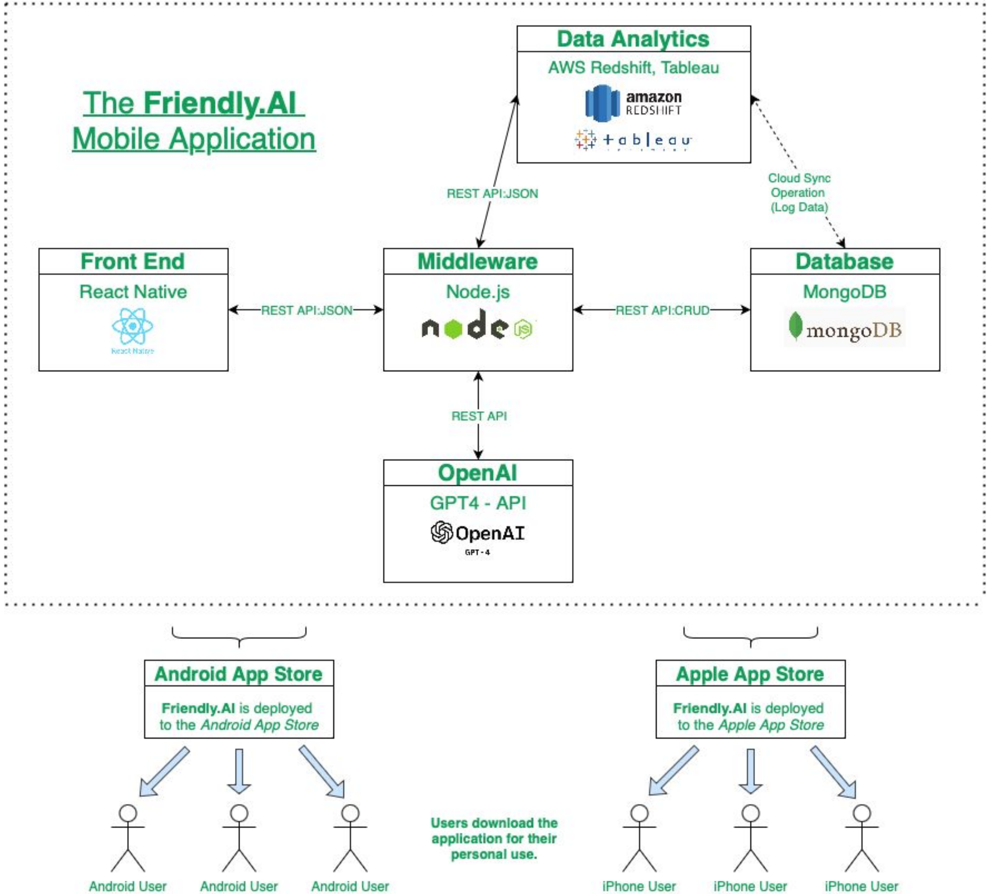
- The technology used is state-of-the-art, with the proliferation of LLMs used in a commercial setting being <1 year in history. Most of the other technologies used for the software development and infrastructure are more mature, but the LLM implementation is truly what runs **Friendly.AI**.

- **Friendly.AI** pushes technological boundaries on the front end when users chat with the AI; we have avatars that are truly outstanding, delivering an incredible experience. The user feels like they are talking to a real friend. The chatbot is not just talking through a prompt; it's talking through **voice over** like you and I would. Therefore, the user is going to know that their friend will ALWAYS be there for them.

3. AI Technology Stack Schematics

The Key Points

Hardware: Android phone, iPhone.
Software: React Native, Node.js, OpenAI GPT4 - APIs, MongoDB, AWS Redshift, Tableau.
Data: 1) Personally Identifiable Information (PII) is stored in MongoDB and synced to AWS,
2) Log data in the form of Key Performance Indicators (KPIs) are displayed in Tableau.
Algorithms: 1) GPT4 pre-trained model, 2) Time series/forecasting graphs, and 3) ML models.



The **Friendly.AI** Mobile Application

Data Analytics
AWS Redshift, Tableau

Front End
React Native

Middleware
Node.js

Database
MongoDB

OpenAI
GPT4 - API

REST API:JSON

REST API:JSON

REST API:CRUD

REST API

Cloud Sync Operation (Log Data)

Android App Store
Friendly.AI is deployed to the *Android App Store*

Apple App Store
Friendly.AI is deployed to the *Apple App Store*

Users download the application for their personal use.

Hardware and Infrastructure

- **Hardware**: Android phone, iPhone
- **Infrastructure**: Amazon Web Services (AWS)

- The hardware that the end user needs to own would be an Android phone/iPhone to download **Friendly.AI** from the Android App Store or the Apple App Store.

- The pipeline that will store users' *personally identifiable information* (*PII*) is to first ingest the data to MongoDB and then the data is synced to a replica on AWS. The data to build dashboards that display *key performance indicators* (*KPI*) will be processed and stored on AWS.

- AWS and OpenAI are of course *commercial-off-the-self* (*COTS*) software providers who have their own hardware that's not under **Friendly.AI**'s control. AWS is an IaaS provider and Open AI is a SaaS provider. Therefore when it comes to hardware, **Friendly.AI** will stipulate that the user owns a mobile phone that's able to download the mobile application.

Software Development

- **Software Used**: React Native, Node.js, MongoDB, OpenAI GPT4 - API, AWS (Redshift), Tableau
- **Programming Language**: HTML, CSS, JavaScript

- With respect to software development, the *frontend* is going to be developed using **React**, the *middleware* and server-side programming is going to be developed using **Node.js**, and the *database* is going to be a NoSQL **MongoDB** server. All communication is done through REST APIs.

- **JavaScript** is going to be used for both the client-side and the server-side programming. A JavaScript library or some third party tool will be used to create the actual avatars that's displayed for the user to chat with.

- The middleware layer is going to incorporate **OpenAI's GPT4 API** that will power the main chatbot functionality, providing the conversation feature that is the core feature of the app.

- Log data in the form of *JSON* will be parsed and stored in **AWS Redshift**. The business intelligence tool **Tableau** is going to be used as the business intelligence tool to keep track of the KPIs that are important to make future business decisions.

Handling Data

- **Database**: MongoDB will be used to store *personally identifiable information* (*PII*)
- **AWS RedShift**: AWS RedShift will be used to store parsed log data that's then displayed in Tableau
- **Business Intelligence**: Tableau, owned by Salesforce, will be used for dashboarding

- By signing up to the **Friendly.AI** application, the user will have to give *PII* such as: first name, last name, phone number, email, date of birth, gender, and age. All this information will be stored on an encrypted Mongo database hosted on the database server and synced to a replica in AWS.

- Once **Friendly.AI** is deployed on a user's mobile device, it's going to generate lots of *logs and metrics*. That data is also going to be stored in AWS. The data is going to be then synced continuously and parsed such that it's stored on AWS RedShift through an AWS Lambda function.

- Once the data (both *PII + logs and metrics*) is in RedShift, the data is now in the **enterprise data warehouse** (**EDW**). With the cloud migration complete, the data is going to be deleted locally on the MongoDB database.

- Business intelligence analysts can now do data mining to mine the business analytics that's important.

Algorithm Development

- **Algorithms**: GPT-4 pre-trained model, time series analysis/forecasting, machine learning modeling

- The core algorithm is the pre-trained *GPT-4 model* that will power the conversational chatbot. GPT-4 will be used initially, but it could be potentially upgraded to the even more potent *GPT-4-32k*. These pre-trained models are of course *large language models* (*LLMs*).

- Log data will be mined to in the form of time series analysis on *key performance indicators*: **1)** average time spent on app, **2)** average logins per day or week for U.S. users segmented by age, **3)** average returning users month-to-month, **4)** churn rate, etc.

- Forecasting and predictive analytics can be used to predict outages and other key application metrics on the **technology side**. Moreover, machine learning modeling will be used to predict future usage rates, customer retention rates, among other user metrics on the **business side**.