A Health Monitoring and Water Consumption Tracking System Based on Smart Cup Using Artificial Intelligence and Internet of Things

Haowen Yu[1], Soroush Mirzaee[2]

[1]Skyline High School, 1122 228th Ave SE, Sammamish, WA 98075
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

Abstract

Dehydration poses a significant global health risk, impacting a diverse population across various age groups [1]. Despite numerous efforts to promote hydration, traditional methods often rely on manual logging in mobile applications, a process prone to user neglect and inaccuracy. SmartFlask, an innovative hydration-monitoring device, seeks to address these limitations by offering a fully automated solution that seamlessly integrates into users' daily routines [2]. Equipped with a time-of-flight (ToF) sensor, SmartFlask tracks water consumption without requiring user input, while an AI algorithm recommends personalized daily water intake based on individual factors such as height and weight [3]. By eliminating the need for manual tracking and providing tailored hydration goals, SmartFlask offers a comprehensive approach to improving hydration habits. This paper details the development, functionality, and potential health benefits of SmartFlask, demonstrating its promise as an effective tool for addressing dehydration on a global scale.

Keywords

Dehydration, ToF, AI algorithm

1. Introduction

This project originated from an assignment in my sales and marketing class in freshman year. We were tasked with making a project for a mock shark tank show, and I came up with the concept of SmartFlask. As I was coming up with the concept, I stumbled upon the statistic that over 70% of Americans, as well as Germans, Australians, and French, suffer from dehydration. Clearly, dehydration was a worldwide problem. After seeing these statistics, I decided to make Smartflask a reality, especially due to the wide range of people dehydration effects. Dehydration can affect all sorts of people ranging from children to old men and women. Dehydration can cause many problems, including but not limited to heart rate problems, blood pressure problems, kidney damage, risk of kidney stones, and muscle damage. Chronic dehydration can lead to a continuation of all the problems listed [4]. Dehydration has continuously been a dire problem the world needs to solve but has not found a solution to just yet.

Many attempts have been made in the past to solve dehydration. Often, these programs involve tracking the water intake of the user by prompting the user to log their water intake in some sort of mobile app [14]. However, more often than not, the user will forget to log their water intake as they do not habitually track their water intake after every sip of water they consume. Furthermore, most users are unaware of the size of their water bottle, and therefore unsure of how much water they are drinking, leading to them being unable to log their water intake without being greatly inconvenienced. With SmartFlask's toF sensor and entirely automatic system, the user does not need to do any logging at all. The sensor will automatically detect their water levels and report the volume of their consumed water accurately. All the user needs to do is to input their water bottle dimensions one time. Furthermore, SmartFlask can also utilize AI to tell the user their recommended water intake amount for their height and weight. Most people do not know their optimal water intake per day, and often end up drinking too much or too little.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. How can the Sensors Stay Accurate and up to Date in Circumstances with No Internet or Service?

The sensors in the SmartFlask system are connected to a boron microcontroller, which connects to a mobile device via cellular network or bluetooth. A bluetooth connection requires no data usage or wifi availability [5]. As for accuracy, the sensors track the amount of water intake by the water level of the water in the water bottle. This is averaged out over time, so even if the water bottle is unstable, the sensors will still reflect accurate information.

2.2. How do the Databases Store all Necessary Data Securely?

All user data will be stored in the cloud via firebase. The data will be directly imported from the boron, and the app will import the data from firebase. Using this system, data can be stored securely in the cloud. Even if the user chooses to log on from various other devices, they will still be able to access their data just the same.

2.3. How does the App Contribute Efficiently to User Experience?

The app contributes to the user experience in a variety of ways. For example, it effectively allows the user to visualize how much water they've drank throughout the day as well as throughout the week, or month, or however long they want to set the timespan into. They can see the data visualized into charts that allows them to see the highs and lows that their water intake amount has experienced. Furthermore, the app also entails an AI chatbot which the user can talk to [6]. This chatbot can recommend water levels that are suitable for the user's height, weight, BMI, and other data that the user can choose to tell the AI. Moreover, there are diet choices in the app that the user can choose from to view the recommended water intake amount for each of these diets. Last but not least, the app can ensure that the user can access their data from any device that they choose to pair the boron microcontroller to, and that the data will stay consistent and sync across all of the devices.

3. SOLUTION

When the user first opens the Smartflask app, they will be greeted with the home page storing all their data for the day as well as the week. In addition, there will also be a button for the user to set a reminder to drink water. On the bottom snackbar, the user can navigate through the explore page, profile page, and insights page. On the explore page, there are multiple features that the user can use to find more about the science of water and the relationship between water and the body. The user can chat with an AI to learn more about the best habits for themselves and how much water they should drink, and they can also click on the different types of diets near the bottom of the page to educate themselves on the types of diets that can be helpful to different types of lifestyles. On the Insights page, the user can toggle the timespan displayed on the chart of their water consumption over time. The toggle ranges from the day of to the entire timespan that the user has used this app. Lastly, on the profile page, the user can view and edit their goals, and perform other basic functions such as signing out and deleting their account.

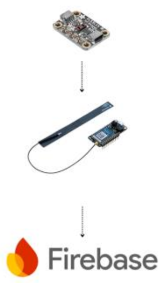
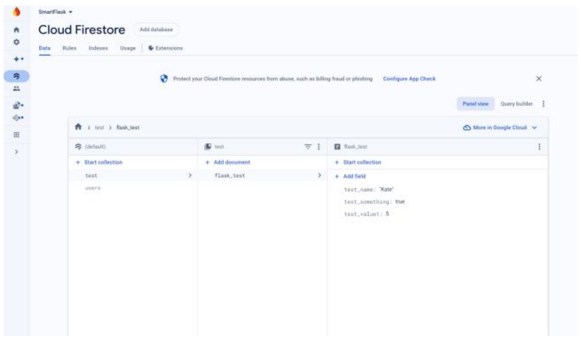


Figure 1. Overview of the solution

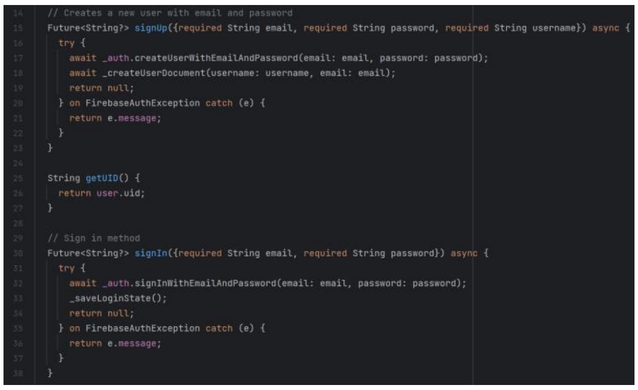
To ensure that all the user data is saved, Firebase authentication is an important part of the authentication document [7]. By incorporating Firebase, all of the user's data is retrieved successfully and ready for the user to access within the app. More importantly, firebase and firestore keeps all of the users data synced through different devices, so the user can choose to login at whichever device they prefer.



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

```
test_name: "Kate"
test_something: true
test_value1: 5
```

Figure 2. Screenshot of test

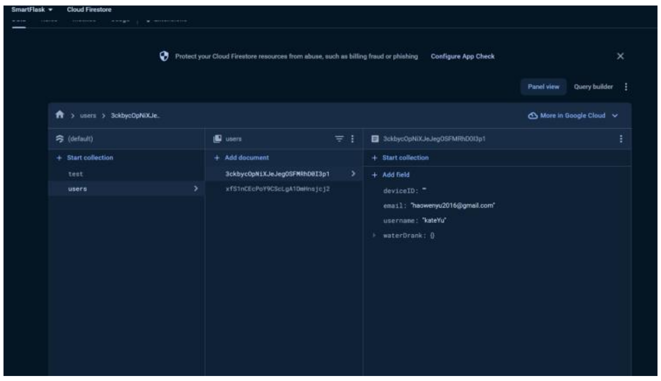


```
// Creates a new user with email and password
Future<String?> signUp({required String email, required String password, required String username}) async {
  try {
    await _auth.createUserWithEmailAndPassword(email: email, password: password);
    await _createUserDocument(username: username, email: email);
    return null;
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}

String getUID() {
  return user.uid;
}

// Sign in method
Future<String?> signIn({required String email, required String password}) async {
  try {
    await _auth.signInWithEmailAndPassword(email: email, password: password);
    _saveLoginState();
    return null;
  } on FirebaseAuthException catch (e) {
    return e.message;
  }
}
```

Figure 3. Screenshot of code 1

This code shows the excerpts describing how the sign in process works with username and emails and password. Using firebase, this information is stored within the cloud database and returns the accurate message, which allows the user to sign in. This code snippet also includes the code for creating new users with email and password for the users that have not signed up before and are new to the app. Furthermore, by using FirebaseAuthenticator, it simplifies the process of creating a new user greatly by calling the createUserWithEmailAndPassword method, and the signInWithEmailAndPassword method.

Smartflask uses a boron microcontroller, lithium battery, and a toF sensor [8]. All of them play different roles in the system of Smartflask. For example, the boron allows the phone to pair successfully to the time of flight sensor and acquire the data. For the time of flight sensor, it will be located on the cap of the water bottle, which will allow it to detect the distance the water level is from the cap, and can therefore calculate the amount of water remaining in the bottle. The lithium battery powers this entire process.



```
deviceID: ""
email: "haowenyu2016@gmail.com"
username: "kateYu"
waterDrank: {}
```

Figure 4. Screenshot of users

In this case, it is not a piece of code but a piece of the firebase console on display. By using firebase to store the data that the time of flight sensor gathers, the data is stored securely and updated regularly on the firebase backend [9]. On many factors on the homepage, such as the indicator of how much water the user has drank today on the homepage and the multiple modes of charts on the data on the statistics page, the data is retrieved from Firebase and updated live to make sure that those elements in the app are up to date with the user's water consumption.

As described in the many parts that come before, the smartflask app seamlessly blends itself with the data and aims to create a streamlined experience for users to view all their data at a glance and be able to view whichever part of their data or account they would like to. The app features many cards and widgets as well as other code to make up charts and diagrams for the user's convenience.

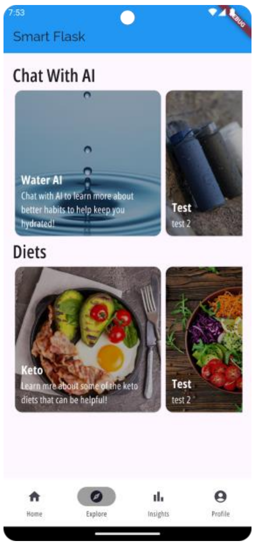
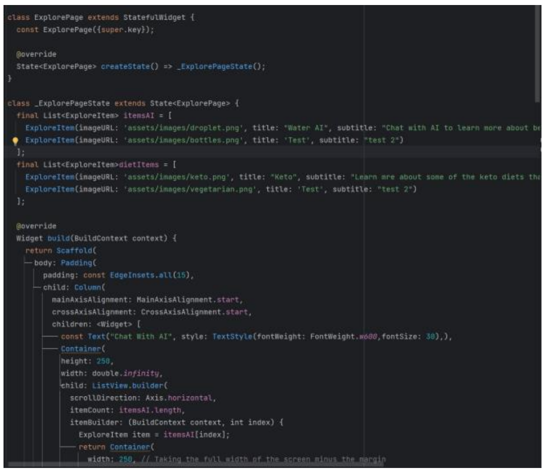

7:53
Smart Flask
Chat With AI
Water AI
Chat with AI to learn more about better habits to help keep you hydrated!
Test
test 2
Diets
Keto
Learn mre about some of the keto diets that can be helpful!
Test
test 2
Home
Explore
Insights
Profile

Figure 5. Screenshot of the app



```
class ExplorePage extends StatefulWidget {
  const ExplorePage({super.key});

  @override
  State<ExplorePage> createState() => _ExplorePageState();
}

class _ExplorePageState extends State<ExplorePage> {
  final List<ExploreItem> itemsAI = [
    ExploreItem(imageURL: 'assets/images/droplet.png', title: "Water AI", subtitle: "Chat with AI to learn more about be
    ExploreItem(imageURL: 'assets/images/bottles.png', title: 'Test', subtitle: "test 2")
  ];
  final List<ExploreItem>dietItems = [
    ExploreItem(imageURL: 'assets/images/keto.png', title: "Keto", subtitle: "Learn mre about some of the keto diets tha
    ExploreItem(imageURL: 'assets/images/vegetarian.png', title: 'Test', subtitle: "test 2")
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Padding(
        padding: const EdgeInsets.all(15),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget> [
            const Text("Chat With AI", style: TextStyle(fontWeight: FontWeight.w600,fontSize: 30),),
            Container(
              height: 250,
              width: double.infinity,
              child: ListView.builder(
                scrollDirection: Axis.horizontal,
                itemCount: itemsAI.length,
                itemBuilder: (BuildContext context, int index) {
                  ExploreItem item = itemsAI[index];
                  return Container(
                    width: 250, // Taking the full width of the screen minus the margin
```

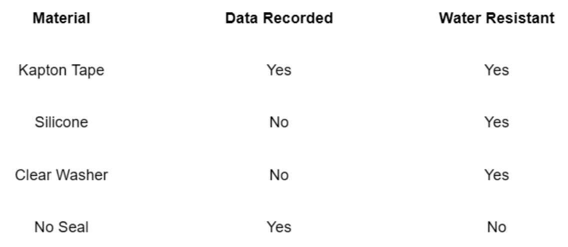
Figure 6. Screenshot of code 2

This code is the code for the previous screenshot shown of the explore page. As pictured, the explore page is made of multiple lists and images, and a scaffold containing a left aligned column of containers including cards. The two rows of cards are completely identical, reflected in the code for their height and width, both featuring code that allows the user to scroll through the cards horizontally if there are too many to be pictured in the screen.

4. EXPERIMENT

One of the possible blind spots of this project can be the water resistance of the sensor and electronics from the water itself to allow for measurements to take place. This part is important to this program because the distance sensor has the major role of calculating the correct amount of water our user has drunk. Hence, it needs to be placed at the top of the waterbottle cap. However, we have faced a difficulty regarding what product can be used in order to not only keep it stable, but also ensure that it is edible/drinking safe.

The experiment involves testing the water resistance and measurement accuracy of the VL54L4CD sensor when exposed to water. The sensor is mounted inside the water bottle's cap, covered with various sealing materials like silicone, Kapton tape, and clear washers to ensure watertight integrity. The setup aims to evaluate the effectiveness of each material in maintaining sensor functionality while preventing water ingress. Control data is sourced from a dry environment where the sensor's distance readings are unaffected by water. By comparing these readings to those taken in the sealed environment, we can assess each material's impact on accuracy.



Material	Data Recorded	Water Resistant
Kapton Tape	Yes	Yes
Silicone	No	Yes
Clear Washer	No	Yes
No Seal	Yes	No

Figure 7. Figure of test result

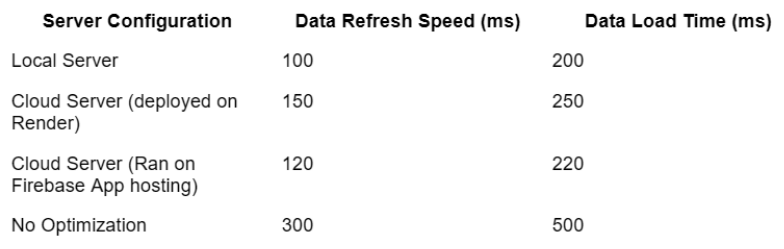
The experiment showed that Kapton tape allowed the sensor to record data, though the readings were occasionally finicky. This suggests that while the tape's thinness and transparency were generally effective, its adhesive properties or application method might have caused sporadic inconsistencies.

In contrast, silicone and clear washers completely prevented the sensor from recording any data. Silicone's thickness and opacity likely obstructed the sensor's ability to function, while the clear washers may have introduced physical barriers or refracted the sensor's signal in a way that blocked accurate readings.

Without any seal, the sensor was able to record data, but this setup left the electronics exposed to potential water interference, highlighting the necessity of a watertight seal.

Overall, while Kapton tape proved to be the most viable solution, further refinements are needed to enhance its consistency and reliability [10]. Future iterations might explore alternative materials or sealing techniques to achieve optimal performance without compromising water resistance.

Another potential blind spot of this project is the data refresh speed within the application. Quick data refresh and loading times are crucial for a seamless user experience.



Server Configuration	Data Refresh Speed (ms)	Data Load Time (ms)
Local Server	100	200
Cloud Server (deployed on Render)	150	250
Cloud Server (Ran on Firebase App hosting)	120	220
No Optimization	300	500

Figure 8. Figure of server results

The experiment revealed that implementing a web server significantly improved data refresh and loading times. The local server configuration provided the fastest refresh speed at 100 ms and a loading time of 200 ms, making it the most responsive option. However, cloud server configurations also performed well, with Cloud Server B showing a slight advantage over Cloud Server A due to better latency management.

Without optimization, the app experienced much slower refresh (300 ms) and load times (500 ms), leading to a suboptimal user experience. The experiment highlights the importance of server setup in enhancing app performance. While local servers offer the best speed, they may not be feasible for all users due to accessibility concerns. Cloud servers provide a good balance between speed and accessibility, making them a viable option for most scenarios. Future improvements might focus on further optimizing server-side code or exploring faster cloud hosting solutions to ensure the best user experience.

5. RELATED WORK

This study, conducted by H.Griffith, features conclusions regarding the accuracy of a sensor in terms of being attached in a water bottle with fluids [11]. The study introduced many numbers and statistics about the improvements in accuracy when the water bottle is placed under different conditions. Some of the crucial differences between this study and SmartFlask is the aim of the project. This project was conducted purely to test the accuracy of the sensors, while SmartFlask was made to remind the user to drink water and regulate water intake of the users. Additionally, this study does not involve the creation of an app with the sensor. As this study only entails a sensor, there is no way for it to connect to a mobile device and carry out any operations without any additional parts. Compared to this study, SmartFlask is much more advanced in terms of allowing the user to monitor and view the data that the sensor has collected from the container

This study, conducted by Dmitry Rodin, involves a hydration sensor that attaches onto the skin of participants [12]. Using photoplethysmographic and galvanic biosensors, this project was tested with participants that were randomly chosen. Each of the participants went through vigorous exercise and medical balance measurements were performed to monitor water mass loss due to perspiration. Compared to SmartFlask, the sensors measure dehydration in a different way. Instead of measuring hydration levels by monitoring water intake, this study monitors dehydration levels by perspiration instead. By conducting the study in this manner, Rodin does not track the water intake of the participants, having no way to acquire the water levels that the participants are drinking. Compared to this, SmartFlask is able to track water levels constantly and tell users if they need to drink more water.

This study, conducted by F. Sabry and T. Eltaras, combines machine learning with hydration monitoring using wearable sensors [13]. The study uses an accelerometer, magnetometer, gyroscope, galvanic skin response sensor, photoplethysmography sensor, temperature sensor, and barometric sensor. Using all these types of data, this study is able to predict the last time a person drank and let the user know when it exceeds a certain threshold. Compared with SmartFlask, Sabry’s study does not track hydration, tracking dehydration instead. Furthermore, it also cannot track total water intake per day. Although reminding the user to drink water when their last time of drinking has passed a long time ago serves as an incentive for the user to drink water, it does not track the amount they drank, therefore being unable to efficiently total up the amount that the user drinks per day nor to know if the user has drank enough. Unlike this study, SmartFlask uses AI to advise the user to intake their ideal water intake based on their weight, diet, and BMI. This study allows the user to personalize the app however they like, which allows for a better user experience.

6. CONCLUSIONS

SmartFlask comes with some minor limitations with its wide variety of features. One of the limitations is that it can only remind the user to drink water using their phone notifications, which sometimes prove to be ineffective, as most users will not leave their notifications on in case of the notifications interrupting something important [15]. If time permitted, it would be ideal to make some sort of screen or a light-up indicator on the surface or cap of the bottle to indicate the progress of the water intake of the user at a glance. Specifically, this indicator could divide the goal of the user's water intake per day by hour, and turn red when the user has lagged behind on their water intake. A couple improvements that could also be made are perhaps beautifying the app a bit more to make it more aesthetically appealing, and add more features such as customizing the colors of the indicator lights or setting custom indicators. For the future, this program can be expanded by creating more SmartFlask bottle caps, and distributing them.

REFERENCES

[1] Thomas, David R., et al. "Understanding clinical dehydration and its treatment." Journal of the American Medical Directors Association 9.5 (2008): 292-301.

[2] Mengistu, Yehenew, et al. "AutoHydrate: A wearable hydration monitoring system." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.

[3] Mamyrin, B. A. "Time-of-flight mass spectrometry (concepts, achievements, and prospects)." International Journal of Mass Spectrometry 206.3 (2001): 251-266.

[4] Armstrong, Lawrence E. "Challenges of linking chronic dehydration and fluid consumption to health outcomes." Nutrition reviews 70.suppl_2 (2012): S121-S127.

[5] Dideles, Myra. "Bluetooth: a technical overview." XRDS: Crossroads, The ACM Magazine for Students 9.4 (2003): 11-18.

[6] Lee, Ju Yoen. "Can an artificial intelligence chatbot be the author of a scholarly article?." Journal of educational evaluation for health professions 20 (2023).

[7] Moroney, Laurence, and Laurence Moroney. "Using authentication in firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 25-50.

[8] Auden, Elizabeth C., et al. "Thermal neutron-induced single-event upsets in microcontrollers containing boron-10." IEEE Transactions on Nuclear Science 67.1 (2019): 29-37.

[9] Sharma, Divya, and Hiren Dand. "Firebase as baas for college android application." International Journal of Computer Applications 178.20 (2019): 1-6.

[10] Benford, D. J., T. J. Powers, and S. H. Moseley. "Thermal conductivity of Kapton tape." Cryogenics 39.1 (1999): 93-95.

[11] Griffith, Henry, Yan Shi, and Subir Biswas. "A container-attachable inertial sensor for real-time hydration tracking." Sensors 19.18 (2019): 4008.

[12] Rodin, Dmitry, et al. "An accurate wearable hydration sensor: Real-world evaluation of practical use." Plos one 17.8 (2022): e0272646.

[13] Sabry, Farida, et al. "Towards on-device dehydration monitoring using machine learning from wearable device's data." Sensors 22.5 (2022): 1887.

[14] Armstrong, Lawrence E., and Evan C. Johnson. "Water intake, water balance, and the elusive daily water requirement." Nutrients 10.12 (2018): 1928.

[15] Pielot, Martin, Karen Church, and Rodrigo De Oliveira. "An in-situ study of mobile phone notifications." Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. 2014.