Smart Beachmat
Thesis Project Proposal

WRITTEN FOR ALEX PUDMENZKY

WRITTEN BY ADAM DOROGI-KAPOSI, 4408496

Abstract

In order to come up with the best approach to tracking our UV exposure, multiple methods have been approach. This report outlines the chosen methods, aims, and progress of the thesis project, and will go into detail which technologies have and will be used during the development.

The project makes use of multiple frameworks in order to create a smartphone application capable of communicating with a UV sensing IoT device. The timeline of the project and associated risks are also discussed.

Table of Contents

1. Introduction

Australia is the leading country for skin cancer (melanoma) rates for both sexes as of 2018 [1]. Australia's ozone layer has thinned by 5-9% since 1960, increasing Australia's risk to exposure of harmful ultraviolet-B (UV-B) radiation from the sun [2].

Since prolonged UV-B radiation has been linked to skin cancer [3], this project aims to raise awareness of such radiation, and give its users the control to track, visualise, and mitigate their exposure. In general, the thesis project consists of a beach mat, with UV sensors, giving the ability to its users to track the UV radiation through a smartphone application.

This project aims to research whether use of such apps link to a reduced rate of skin cancer.

This project aims to prevent skin cancer by giving the users the guidance and knowledge of tracking their UV radiation exposure, alerting users of dangerous levels of sun exposure. This will be done based on the users skin type, skin conditions, gender, and age (more precise calculations for skin conditions). The UV radiation will be calculated based on a combination of visible and infrared (IR) light.

This thesis will cover and contribute to the development of hardware, firmware, and client and server side software, to provide a reliable, user friendly product. This aims to help people all around the world keep track of their sun exposure when at the beach, using the smart beach mat. A more detailed project outline will be covered in further sections.

2. Background

2.1. Relevant Technologies

2.1.1. Bluetooth Low Energy

Bluetooth Low Energy (BLE), is a wireless communication technology used for short range communication between BLE compatible devices, such as a mobile phone, and BLE enabled blood pressure monitors. Bluetooth operates in the 2400-2483.5 MHz range in the 2.4 GHz Industrial, Scientific and Medical (ISM) radio frequency band [4]. BLE's low power consumption allows devices to run on a small batter for 4-5 years for devices that need to exchange small amounts of data periodically [4]. BLE's short range communication allows high transfer rate of data of 1 Mb/s, making it suitable for IoT devices [4].

2.1.2. Hypertext Transfer Protocol

Similarly to BLE, Hypertext Transfer Protocol (HTTP) is a standardised communication protocol that can also be used to wirelessly transmit data between a client and a server, such as a mobile phone and a web server, over the internet [5]. Unlike BLE however, HTTP is used to communicate over a long range, such as between countries, and is done to a web server. The HTTP protocol is made up of HTTP requests, such as a GET, POST, PUT, PATCH, or DELETE request, which can instruct the web server to create, read, update, and delete contents of a web page [6]. The HTTP response is the response that is sent back to the client, following the client's HTTP request, to indicate the status of the request with a status code, and optionally return data to the client [6].

2.1.3. Flutter

Flutter is a framework for the Dart programming language developed by Google in 2017 [7]. It can be used to create beautiful, cross-platform mobile applications in record time [8], allowing fast development, expressive and flexible UI, and native performance. Since Dart is a functional programming language, and with the use of Flutter, introduces a "hot reload" feature (stateful hot reload). This preserves the state of your application, enabling you to view the effect of changed code almost immediately, without having to completely rebuild the mobile application [9]. This may save countless hours of development time in the long run.

2.1.4. PHP

PHP is a scripting language, often used on the server-side, such as a web server. It can be used for web development, for example, by generating dynamic HTML pages on the server when the client makes a HTTP request, and returning that HTMl page to the client. PHP can also be used to connect to databases, and perform create, read, update, and delete (CRUD) operations on databases. Due to this feature, PHP is often used to generate HTML pages based on contents of databases, such as displaying a list of users on a web page [10]. In other words, PHP is used to link the client HTTP request to a database operation, optionally returning a database record to the client application. PHP can accept any client HTTP request, and it doesn't necessarily have to come from a webpage. The client request can also come from other applications, such as desktop applications, smartphone applications, or IoT devices.

2.1.5. MySQL

MySQL is an open-source Structured Query Language (SQL) database management system. It is a relational database, meaning the relationship between each entity in the database is defined during the construction of the database, storing the data in tables, where each row represents an instance of data, and the column names represent the different variables for that data [11]. MySQL is often used for web applications, to store information specific to the web application, such as usernames, passwords, or products in an online store. These database entries can be created, retrieved, updated, and deleted either manually, or by using a server-side programming language, such as PHP, as described in Section 2.1.4.

2.2. Related Works

2.2.1. Shade

Shade is an ultra-accurate clinical-grade wearable UV sensor, backed by the National Cancer Institute, and the National Science Foundation [12]. This wearable piece of technology, clipped to clothing by magnets, empowers anyone to proactively manage sun exposure, and is proved to be 5x-30x more accurate than other UV wearable according to their publication *"A Comparative Study of Wearable Ultraviolet Radiometers"* [13].

The paper outlines the challenges and motivations of developing such a product, and compares the product's results to other wearable UV sensors, based on their accuracy, and sensitivity. The paper compares multiple varieties and classes of UV sensors, which can be seen in Table 1. Due to the lack of standardisation for the accuracy of such devices, the paper's research has decided to adapt a research-grade instrument (X1-4 UVE radiometer) to provide a reference UV index measurement.

TABLE 1 — PERFORMANCE OF COMMERCIAL UV SENSORS.

Product (Company)	Measure UVB	Measure UVA	Rejects Visible
X1-4 UVE radiometer (Gigahertz)	✓	✓	✓
Shade (Shade)	✓	✓	✓
Band (Microsoft)	✗	✗	✓
June (Netatmo)	✓	✗	✓
Sunfriend (Sunfriend)	—	—	✗
UV Watch (Dakota)	✗	✗	✓
Sunsprite (Goodlux)	✗	✗	✓
EPA UVIndex App	—	—	—

Shade's research paper has also compared the sensors based on their price, form factor, sensor category (radiometer or dosimeter), and their IV index resolution.

The paper outlines 162 measurements on each device, under open New York skies, covering a UV index range of 0 to 8. The accuracy of the devices were then determined by the percentage of the measurements being within 30% and 20% of the reference measurements. This result can be seen in Figure 1.

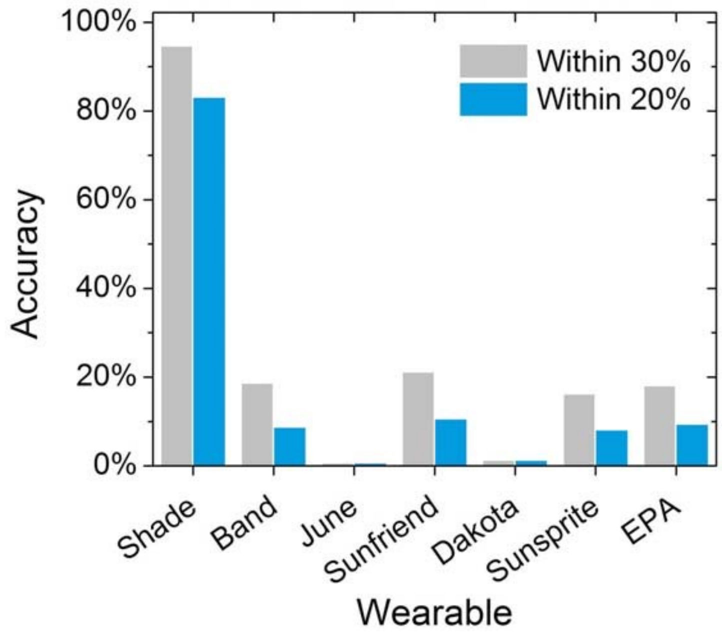


FIGURE 1 — ACCURACY OF COMMERCIAL UV SENSORS, WITH THE X1-4 UVE RADIOMETER USED AS REFERENCE [13].

The Shade mobile application accounts for people's differences in UV limits and sensitivity, taking into account their skin type, medication, age and history of prior sunburns, as even with people with the same skin type often have large differences in UV exposure sensitivity [14]. Shade's application addresses this problem by recommending UV limits during the initial on-boarding screen of the application, seen in Figure 2. The app also allows tracking of the percentage of daily limit reached, along with the real time UV index, as seen in Figure 3. The full sized images can be found in Appendix A.

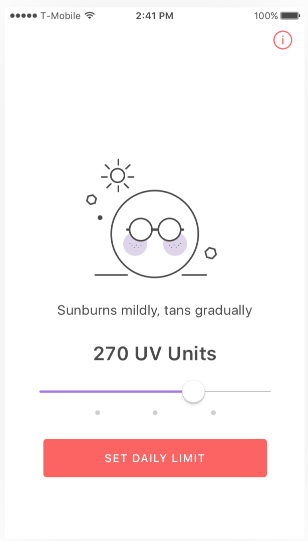


FIGURE 2 — SCREENSHOT OF ON-BOARDING SCREEN IN SHADE'S IOS APPLICATION.

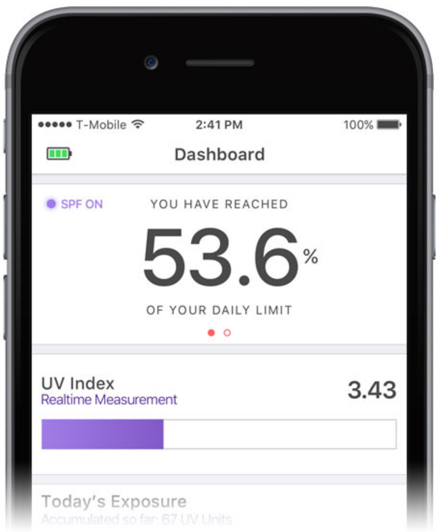


FIGURE 3 — SCREENSHOT OF TRACKING SCREEN IN SHADE'S IOS APPLICATION.

The Shade sensor provides an easy to use, accurate, splash-proof UV measurement device, with a rechargeable battery via a micro USB cable, for a price of US$299 [15].

2.2.2. QSun

The QSun wearable is a Federal Communications Commission (FCC) approved UV exposure tracker, aiming to reduce the risk of sunburn and skin damage, by monitoring UV exposure levels, along with vitamin D levels [16]. The 2018 model of the QSun wearable, along with the QSun smartphone application, provides:

- tracking of sun exposure;
- tracking of vitamin D production;
- personalised tips based on the real time UV index;
- personalised time estimation of sunburn;
- sunscreen amount estimation;
- sunscreen reminders;
- daily UV index forecast;
- UV index map;
- skin health analysis; and

- sun exposure history [17].

The sunscreen recommendation, real time UV index, and sunscreen amount estimation screens of the application can be seen in Figure 4, with a full version in Appendix B.

FIGURE 4 — SCREENSHOTS OF THE SUNSCREEN RECOMMENDATIONS (LEFT), REAL TIME UV INDEX (MIDDLE), AND SUNSCREEN AMOUNT ESTIMATION (RIGHT) SCREENS IN QSUN'S IOS APPLICATION [9].

QSun claims the application communicates/syncs with the wearable via Bluetooth Low Energy (BLE), with an accurate UVA/UVB index reading within +- 0.5 UVI, with a UV range of 0 to 11+, seems to measure both UVA and UVB, similar to Shade in Table 1.

The QSun website [9] also indicates that the wearable is splash proof, and has a long battery life, with a replaceable coin-cell battery (CR2032). The wearable has a 256-kb flash memory, which is capable of saving 45 days of detailed data. The device has an optimum operating temperature of -30C to +55, and a retail price of US$149.

2.2.3. Comparison

In comparison, both Shade and QSun sense both UVA and UVB radiation. UVA and UVB sensors provide more accurate information, which can be used to add extensive features to a product, such as tracking the vitamin D production, since the vitamin D is produced when UVB rays interact with our skin [18]. However, this thesis project will be similar to Microsoft's Band, as seen in Table 1, as it accurately calculates the UV index based on visible and IR light.

Both products rely on a battery, either by recharging or replacing, which poses an inconvenience to the user experience. This project aims to resolve this issue without using batteries that require manual recharging or changing, discussed in more detail in further sections.

Shade and QSun are both splash-proof, which is likely due to their method of recharging their devices. Devices with external ports, such as a micro-USB port, tend to be harder to waterproof, as more openings need careful attention for concealment. The design of this thesis also overcomes this limitation, allowing a potentially fully water, dust, and sand resistant UV index sensor, suitable for beach conditions.

Neither companies have shown evidence to support skin cancer research, such as finding links between the use of their app, and skin cancer rates. This thesis project will link long term UV exposure to the health of their users.

3. Product Design

3.1. Sensors and Data Acquisition

The Smart Beachmat thesis project will use a single UV index sensor; the Adafruit SI1145. However, this sensor does not directly sense UVA and UVB radiation waves. It does not contain a UV sensing element, however, it has a calibrated light sensing algorithm which estimates the UV index based on visible and infra red light [19].

The SI1145 is a low powered sensor, and can be configured to include features such as proximity sensing, and ambient light sensing. To sense the visible and IR light, it is integrated with high-sensitivity visible and infrared photodiodes [20]. It offers outstanding performance under a wide range of light sources, such as sunlight, and even under dark glass covers. It also offers data transmission using I2C Serial communications, with up to a 3.4 Mb/s transfer rate. This sensor is also capable of operating from 1.71 to 3.6 V within -40C and +85C temperatures [20].

This high upper limit on the operating temperature makes it suitable for keeping under warm conditions for longer periods of time, such as direct sunlight, which complements its main use case. The UV sensor runs on 3-5 volts [21].

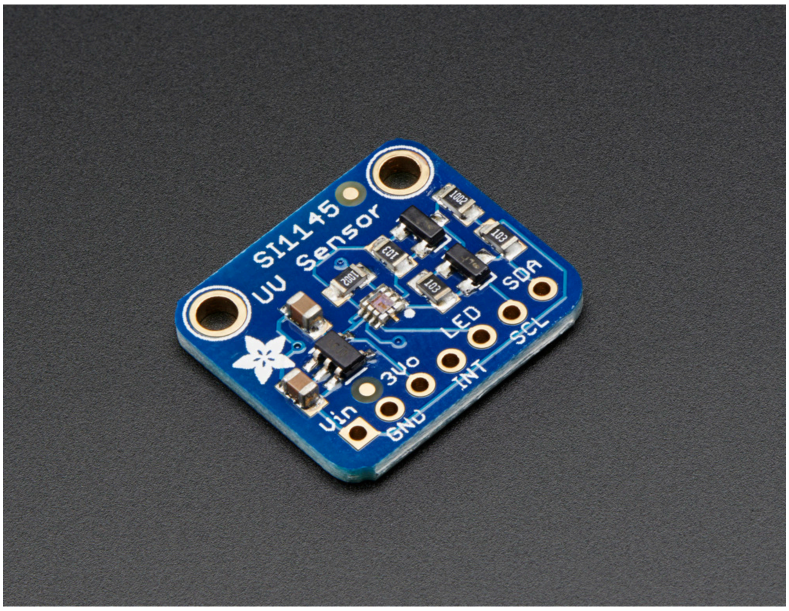


FIGURE 5 — THE ADAFRUIT SI1145 VISIBLE LIGHT AND IR SENSOR [19].

3.2. Micro-controller

The data from the SI1145 sensor will be transmitted an ESP32 micro-controller. This micro-controller is responsible for receiving the data through the sensors I2C interface, and forwarding this data to a mobile application using the micro-controller's BLE interface. The ESP32 is also capable of WiFi and BLE communication. Two versions of the ESP32 micro-controller will be used throughout the development of the project:

- ESP32 Thing development board; a developer friendly development board, with components such as micro-USB ports, LEDs, and through-hole pins included. This will be used for prototyping the initial version of the project, as it simplifies the development process.

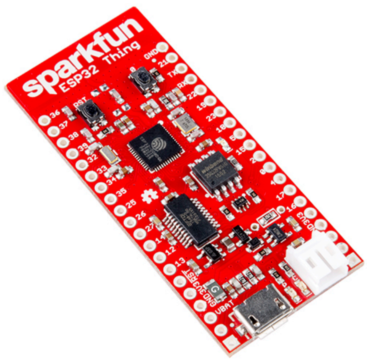


FIGURE 6 — THE ESP32 THING DEVELOPMENT BOARD [22].

- ESP32 Wroom chip; a production-ready, minified version of the ESP32 Thing development board. This is only the chip, and therefore does not contain any additional components, such as LEDs, or USB ports.

FIGURE 7 — THE ESP32 WROOM CHIP [23].

Although two different versions of the ESP32 will be used throughout development, their core functionality remains the same. The ESP32 is capable of withstanding temperatures ranging from -40C to +150C, however, the recommended operating temperature is to be between -40C to +85C [24]. This makes it extremely suitable for exposure to higher temperatures, however, further mitigation will be required in order to make it suitable for operating in direct sunlight for longer periods, as discussed in the project plan section. The ESP32 runs on 2.2-3.6 volts [22].

3.3. Power Supply

One of the main drawbacks of the compared models in Section 2.2 was their battery life, and modes of battery recharging. These included recharging the UV sensor through a micro-USB cable, or replacing a cell-coin battery. None of these methods take advantage of the energy provided from the sun.

This thesis project will rely on solar power to power the ESP32, along with a rechargeable battery. However, the battery will be recharged by the solar panel, therefore it will not need manual recharging. The solar cell will both power the ESP32 and charge the battery while in sunlight. If there is low to no sunlight, the ESP32 will rely on the battery for power.

This method of providing energy also allows full encapsulation of the device, without relying on any external ports, this allowing the device to be completely waterproof.

Since the UV sensor can take anywhere between 3-5 volts, and the ESP32 takes 2.2-3.6 volts, we'll need a battery that can provide anywhere from 3 to 3.6 volts, so it can successfully power both the UV sensor and the ESP32. A lithium-ion batter will be used for this purpose, due to its great energy density, close to three times that of Nickel-Cadmium batteries. Lithium is the lightest known metal, providing the largest energy density per weight [25].

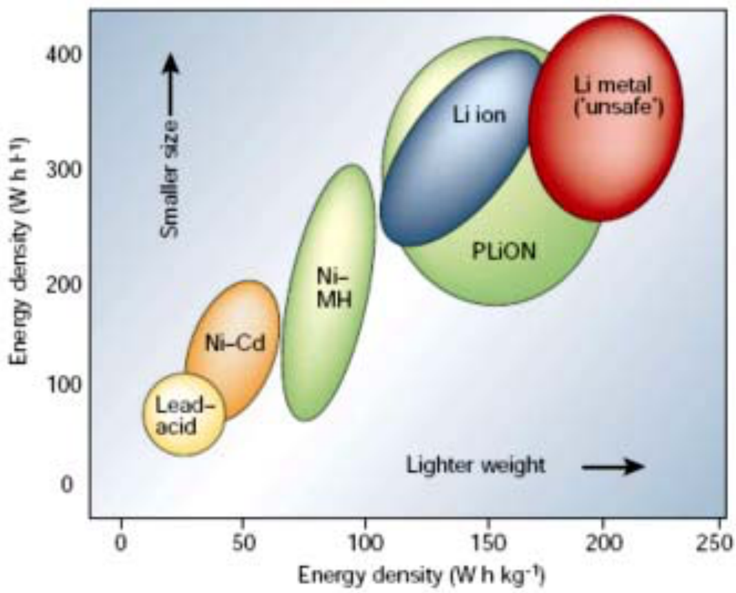


FIGURE 8 — ENERGY DENSITY OF BATTERIES BY SIZE AND WEIGHT [26].

The solar power will work by charging the battery, and the related components (ESP32, UV sensor) will draw power from the battery. In order for the solar power to safely and efficiently power the battery, a solar charge controller will be needed. This will regulate the voltage and current, keep batteries from overcharging, and ensure no current flows back to the solar panel, hence draining the battery [27].

The basic setup of a solar charge controller can be seen in Figure 9.

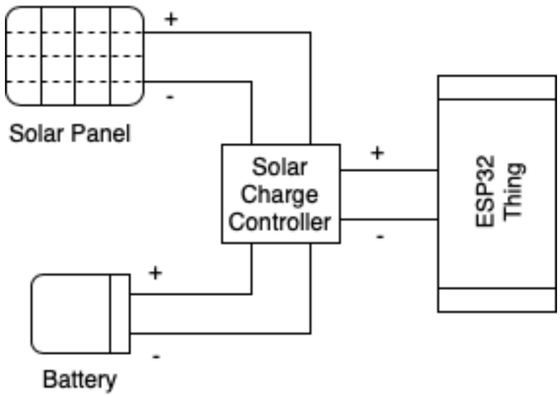


FIGURE 9 — SOLAR CHARGE CONTROLLER SETUP WITH ESP32.

3.4. Smartphone Application

The smartphone app of this project will be the main form of user interaction for reading the UV indexes, viewing history, and managing user data. Prototype screens of the app can be seen in Figure 10. For the full sized prototypes, see Appendix C.

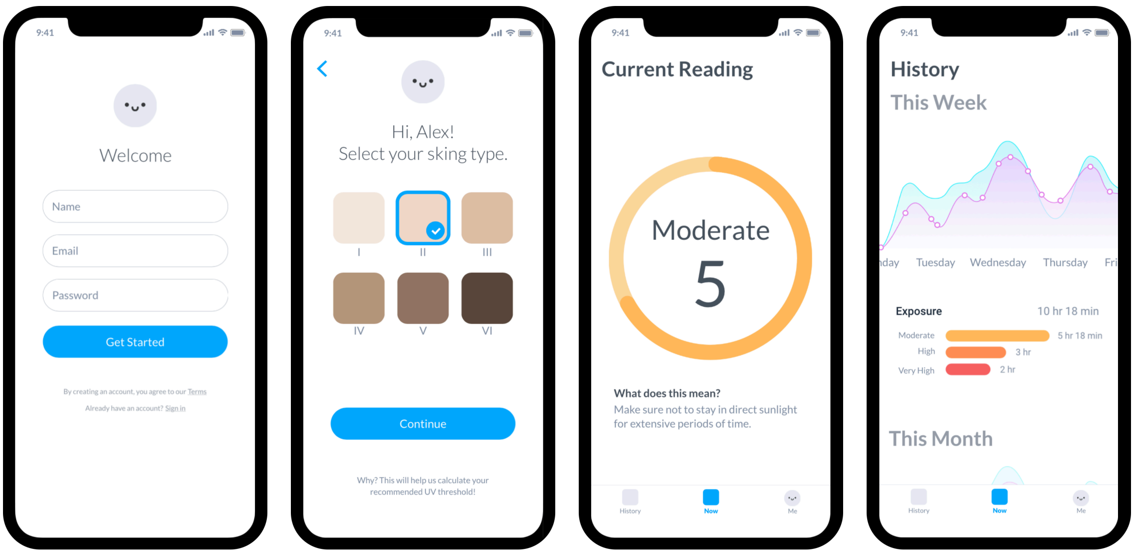


FIGURE 10 — DESIGN PROTOTYPE OF SMARTPHONE APPLICATION FOR SMART BEACHMAT.

The smartphone application will be written in the Dart programming language, using the Flutter framework, to allow for quick development across multiple mobile platforms. The smartphone application will be developed with a clean design, and easy user experience in mind, to make it suitable for people of all ages.

3.5. Communication Methods

In order for the devices to communicate, and to persist valuable information, the following communication protocols will be used:

- Inter-Integrated Circuit (I^2C)
- Bluetooth Low Energy (BLE)
- HTTP

3.5.1. I^2C

When the Adafruit SI1145 reads the UV index based on the visible and infrared light emitted from the sun, it will need to communicate the read UV index to the ESP32. This will be done through the I^2C communication method. I2C is a synchronous, serial, wired communication method, intended for short range communication between master and slave devices [28]. In this case, our master node is the ESP32, as it generates the clock and initiates communication with the slave. Our slave node is the Adafruit SI1145 sensor, which receives the clock and responds to the ESP32 when instructed.

3.5.2. BLE

BLE will be used to communicate the received UV indexes from the ESP32 to the smartphone application, as well as the battery level of the ESP32. In order for the two devices to communicate, they must first establish the communication by first advertising. The device performing the advertising is known as the peripheral, and in this case, is the ESP32 [https://blog.bluetooth.com/a-developers-guide-to-bluetooth]. The device doing the scanning (searching) for advertisements is known as the central device, and will be the smartphone in this case. When performing the advertising, the peripheral device sends advertising packets, which can contain information about that device, such as its name, or service UUID. This way, when the central device is scanning for peripherals, it can filter out the unwanted devices [29]. BLE will also be used to send firmware updates from the phone firmware updates from the phone to the ESP32 in the background, without the user noticing. This is done so the firmware of the ESP32 can be updated with features such as security improvements, without the user having to manually reprogram the device for each update.

A BLE peripheral has a Generic Attributes Profile (GATT), which specify the structure and format of the data exchanged, defining basic elements such as *services* or *characteristics* [30]. A BLE peripheral has one or more services, and each of those services has one or more characteristics. A service is a collection of data and behaviours which define a particular function or feature of the device [30]. These services are standardised by the Bluetooth Special Interest Group (SIG). The services used for this project will be the Battery Service and Environmental Sensing service, with service numbers 0x180F and 0x181A respectively [31]. A characteristic is a value used in a service with information about how the value's format — how the value is represented and accessed [32]. Characteristics are also standardised by Bluetooth SIG. Within the Battery Service, the Battery Level characteristic will be used in order to transmit the ESP32's battery level [33]. Within the Environmental Sensing service, the UV Index characteristic will be used to represent the UV indexes read from the sensor [34].

Once the advertising has been done and the devices are connected, the peripheral will send periodic notifications to the central device. Sending notifications means the phone doesn't have to keep reading the value from the peripheral, as it will be notified by the peripheral when the data is sent. This way, the ESP32 can read the data periodically, and send the data along with a notification to the smartphone.

3.5.3. HTTP

HTTP will be made use of in order to communicate data between the smartphone and the web server and MySQL database. For example, when a user registers to the smartphone application, a user will need to be created in the database on the cloud, so the user's data can be saved and accessed. The smartphone application will also send other data to the server, such as the UV index history of the user, and the location of the user's phone. The location would be needed in order to collect location-based UV information, which may be made publicly available to third-parties, such as the government, as well as other users, so they can look at a map, and see the UV indexes by location.

The representational state transfer (REST) API will be created to aid this client-server communication, and provide a reliable and robust data exchange. The REST API is a standardised software architecture style which defines a set of constraints and formats to send and receive HTTP requests [35]. This standard allows a reusable and robust way of handling and making HTTP requests in order to perform CRUD operations on the server-side.

3.5.4. System Overview

The system architecture will be made up of the combinations of communication methods outlined previously. A system architecture diagram of the used protocols and communication methods can be seen in Figure 11.

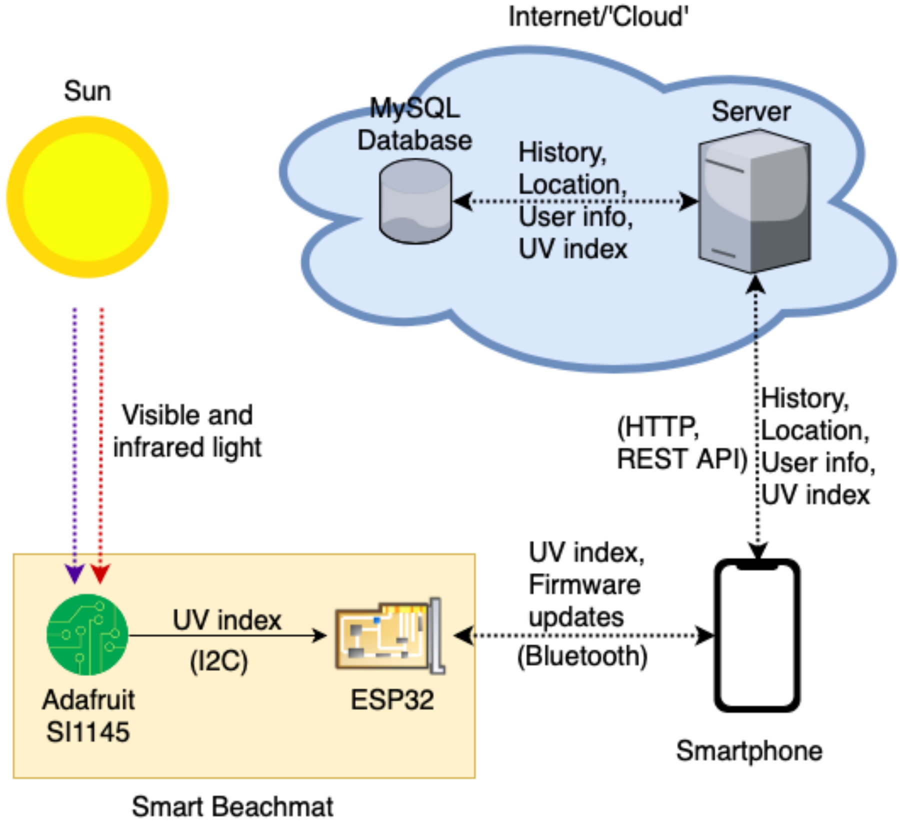


FIGURE 11 — SYSTEM ARCHITECTURE DIAGRAM.

4. Project Plan

4.1. Methodology

The project will be undertaken using the Scrum software development process. Scrum is an Agile framework, where development on a product is undertaken in iterated *sprints*, lasting from 2-4 weeks. Scrum is made up of the following phases and components [36]:

- Product backlog
    - The set of product features and requirements specified by the client (product owner).
- Sprint backlog
    - A subset of the product backlog, specifying which the goals of the upcoming sprint, and what features will be implemented in the product during that sprint.
- Daily scrum
    - A stand-up meeting, where each developing team member has a discussion every day regarding the progress of the sprint. Teach team member will share and discuss their progress since the last stand-up, what they're planning to work on up until the following stand-up, and any blocks or issues they're facing.
- Sprint review
    - A meeting with the client reviews the team's progress, and the product backlog, making adjustments to the plan by adding or removing certain features, to meet the client's needs [37].
- Sprint retrospective
    - The sprint retrospective is a reflection amongst developers in order to determine what went well during the sprint, what could have been done better, and what will be done next. The sprint retrospective is done in order to optimise the team's development process [38].

A diagram of the Scrum process can be seen in Figure 12.

SCRUM FRAMEWORK

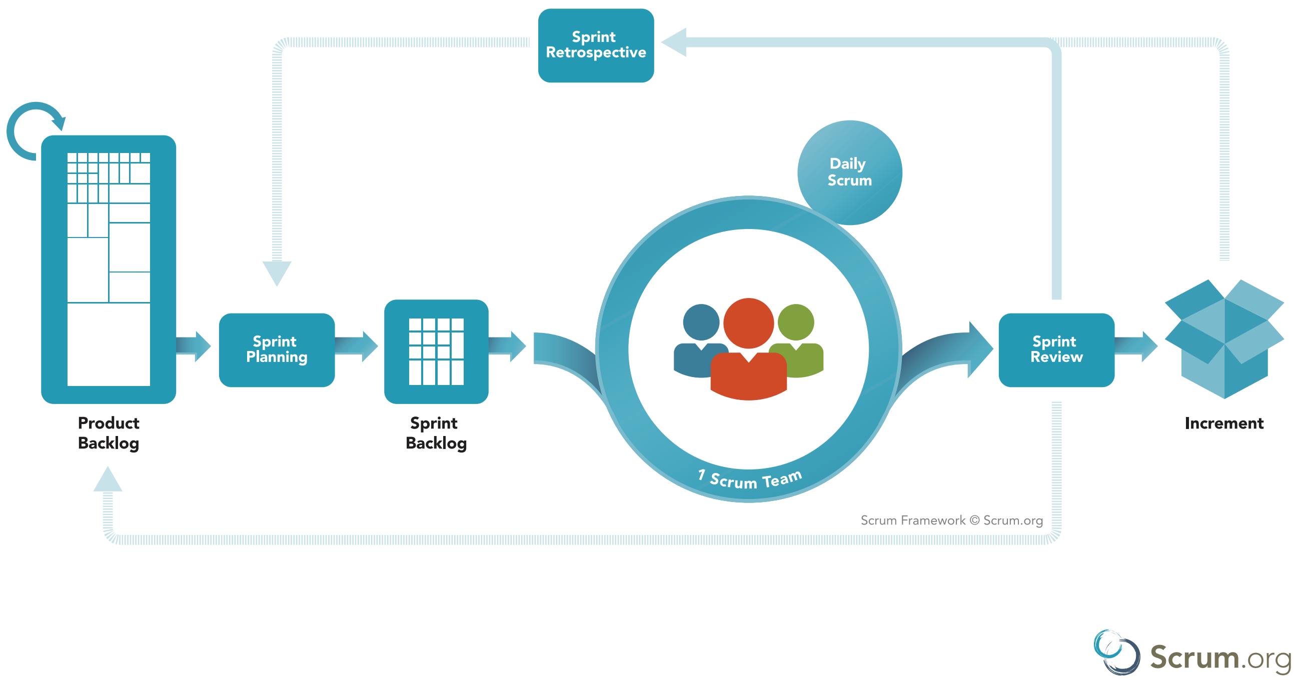


FIGURE 12 — SCRUM FRAMEWORK WORKFLOW [36].

The development team for this project will consist of a single developer, who will undertake 2-week sprints, meeting with the client, and making use of each step of the Scrum framework.

4.2. Code Management

In order to manage the code for the firmware, mobile application, and server, the Git version control will be used. This will ensure that the source code of all components is stored on a central Git server, accessible from anywhere. This will also make it easy to manage code, test out new features, and roll back to certain commits, given an unwanted change.

This will ensure the code is safe and less prone to cause damage from mistakes such as accidentally breaking code, providing flexibility in development.

4.3. Progress

So far, progress has been made regarding the I^{2}C and Bluetooth communications. Some soldering has been done to connect the Adafruit SI1145 sensor to the ESP32, and successfully reading and sending UV indexes to the ESP32 micro-controller. The BLE interface has also been activated on the ESP32, and the ESP32 is capable of notifying a smartphone device, sending data to that device periodically.

The BLE communication has been tested out on an iPhone, with a free application from the AppStore, intended for Bluetooth testing purposes.

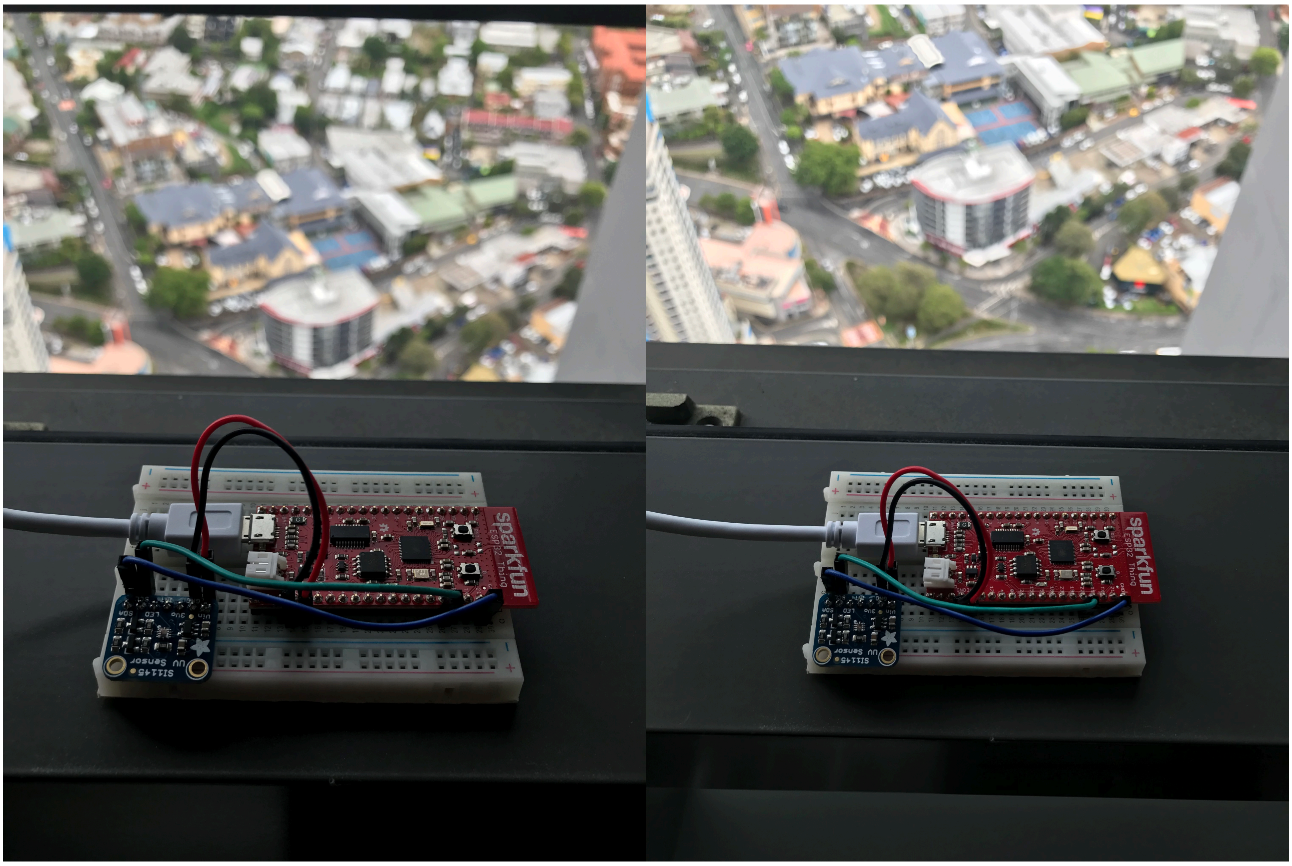


FIGURE 13 — WIRING OF THE ESP32 AND ADAFRUIT SI1145.

4.4. Roadmap

The next steps regarding project development are as follows:

1. *Further testing regarding BLE communication, BLE advertising packets, and BLE device discovery.*
   - Approximate date: March 28, 2019
   - Approximate duration: 1 week
2. *Further research and testing of Lithium-Ion batteries with the ESP32.*
   - Approximate date: April 7, 2019
   - Approximate duration: 2 weeks
3. *Further research and testing of solar panel and solar charge controller.*
   - Approximate date: April 21, 2019

- Approximate duration: 3 weeks

4. *PCB design for ESP32 Wroom, ordering of necessary PCB components.*
   - Approximate date: May 14, 2019
   - Approximate duration: 1 week
5. *Soldering of PCB and components.*
   - Approximate date: May 21, 2019
   - Approximate duration: 1 week
6. *Soldering of PCB and components.*
   - Approximate date: June 1, 2019
   - Approximate duration: 1 week
7. *Design and 3D printing of device casing.*
   - Approximate date: June 7, 2019
   - Approximate duration: 1 week
8. *Further research and testing of device durability, regarding heat dissipation and waterproofing.*
   - Approximate date: June 14, 2019
   - Approximate duration: 1 week
9. *Client-side development of mobile application.*
   - Approximate date: June 21, 2019
   - Approximate duration: 1 month
10. *Database setup and configuration.*
    - Approximate date: July 21, 2019
    - Approximate duration: 1 week
11. *Server-side development of REST API.*
    - Approximate date: August 1, 2019

- Approximate duration: 1 month

12. *Research and testing regarding beachmat material, durability, use case scenarios. Combining the physical device with the smart beachmat.*

- Approximate date: September 1, 2019
- Approximate duration: 2 weeks

Approximate project duration: 22 weeks

Approximate project end date: September 14, 2019

5. Risk Assessment

Every project comes with certain risks, and each of those risks has a certain likelihood and severity. The risk matrix for this project — outlining the risk factors based on risk likelihood and severity — can be seen in Table 2.

TABLE 2 — RISK MATRIX.

			Severity			
		1	2	3	4	5
		Negligible	Minor	Moderate	Significant	Severe
	5 Very Likely	6	7	8	9	10
Likelihood	4 Likely	5	6	7	8	9
	3 Possible	4	5	6	7	8
	2 Unlikely	3	4	5	6	7
	1 Very Unlikely	2	3	4	5	6

The risks associated with this project can be seen in Table 3, along with their risk factor from Table 2. A risk mitigation has also been included to provide tips and steps on how to mitigate the risk, along with the updated risk factors after mitigation.

TABLE 3 — RISK MITIGATION PLAN.

Risk	Risk Factor	Risk Mitigation	Mitigated Risk Factor
Security issues regarding firmware and software, such as stolen user credentials, unauthorised database operations	8	Ensure passwords in database are hashed, ensure encryption of all sensitive data, ensure following security guidelines while developing with Bluetooth and Flutter, account for and mitigate web server attacks through penetration testing	6
Increase skin cancer in users as a result of using this application	8	Restrict application usage if person is exposed to high UV indexes for prolonged periods	6
Electrocution during project development	7	Following laboratory safety rules and guidelines, wearing correct PPE	6

Risk	Risk Factor	Risk Mitigation	Mitigated Risk Factor
Back injury or malformed posture during prolonged code development	7	Take regular breaks, ensure using correct posture, ensure table and chair is adjusted correctly	5
Late shipping of project components	6	Ensure project components are ordered in advance with room for late deliveries	4
Eye strain during prolonged periods of code development	6	Take regular breaks, follow the 20-20-20 rule: every 20 minutes, look at something for 20 seconds, that's 20 feet away.	4
Safety issues regarding hardware, such as explosion or burning of hardware due to overheating, electrocution due to failed waterproofing	6	Continuously test the hardware under variable conditions, accounting for all use cases, and ensure it is robust and safe to use	4
Exceeding project budget	5	Look for the cheapest alternative of components, plan finances in advance	4

References

[1]Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global Cancer Statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin, in press. The online GLOBOCAN 2018 database is accessible at http://gco.iarc.fr/, as part of IARC's Global Cancer Observatory.

[2]"Department of the Environment and Energy", *Department of the Environment and Energy*. [Online]. Available: https://www.environment.gov.au/protection/ozone/ozone-science/ozone-layer. [Accessed: 28- Mar- 2019].

[3]"Twenty Questions and Answers About the Ozone Layer: 2014 Update", *Wmo.int*, 2014. [Online]. Available: http://www.wmo.int/pages/prog/arep/gaw/ozone_2014/documents/2014%20Twenty%20Questions_Final.pdf. [Accessed: 28- Mar- 2019].

[4]B. Ray, "Bluetooth Vs. Bluetooth Low Energy: What's The Difference?", *Link-labs.com*. [Online]. Available: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy. [Accessed: 28- Mar- 2019].

[5]"What is HTTP", *W3schools.com*. [Online]. Available: https://www.w3schools.com/whatis/whatis_http.asp. [Accessed: 28- Mar- 2019].

[6]R. Shuler, "How Does the Internet Work?", *Web.stanford.edu*, 2002. [Online]. Available: https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm. [Accessed: 28- Mar- 2019].

[7]Chris Bracken. "Release v0.0.6: Rev alpha branch version to 0.0.6, flutter 0.0.26 (#10010) · flutter/flutter". GitHub. [Accessed 08- Aug- 2018].

[8]"Flutter - Beautiful native apps in record time", *Flutter.dev*. [Online]. Available: https://flutter.dev. [Accessed: 28- Mar- 2019].

[9]"Hot reload", *Flutter.dev*. [Online]. Available: https://flutter.dev/docs/development/tools/hot-reload. [Accessed: 28- Mar- 2019].

[10]"PHP: What is PHP? - Manual", *Php.net*. [Online]. Available: http://php.net/manual/en/intro-whatis.php. [Accessed: 28- Mar- 2019].

[11]"MySQL :: MySQL 8.0 Reference Manual :: 1.3.1 What is MySQL?", *Dev.mysql.com*. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html. [Accessed: 28- Mar- 2019].

[12]"Shade UV sensor - Funded by the National Cancer Institute", *Wearshade.com*. [Online]. Available: https://www.wearshade.com. [Accessed: 28- Mar- 2019].

[13]S. Banerjee, "A Comparative Study of Wearable Ultraviolet Radiometers", *Storage.googleapis.com*. [Online]. Available: https://storage.googleapis.com/emmanuel-public/Shade%20IEEE%20published.pdf. [Accessed: 28- Mar- 2019].

[14]"Individual sensitivity to UV exposure", *Wearshade.com*. [Online]. Available: https://www.wearshade.com/articles/individual-sensitivity-to-uv-exposure. [Accessed: 28- Mar- 2019].

[15]"Shade UV sensor store", *Shade*. [Online]. Available: https://store.wearshade.com. [Accessed: 28- Mar- 2019].

[16]"QSun | Wearable Sun Tracker | Balance UV exposure and vitamin D", *QSun | Wearable Sun Tracker | Balance UV exposure and vitamin D*. [Online]. Available: https://qsun.co. [Accessed: 28- Mar- 2019].

[17]"Features to Optimize Skin Health | QSun - Sun Safety Advisor", *QSun | Wearable Sun Tracker | Balance UV exposure and vitamin D*. [Online]. Available: https://qsun.co/features/. [Accessed: 28- Mar- 2019].

[18]"Make Vitamin D, Not UV, a Priority - SkinCancer.org", *Skincancer.org*. [Online]. Available: https://www.skincancer.org/healthy-lifestyle/vitamin-d/make-vitamin-d-not-uv-a-priority. [Accessed: 28- Mar- 2019].

[19]"Overview | Adafruit SI1145 Breakout Board - UV index / IR / Visible Sensor | Adafruit Learning System", *Learn.adafruit.com*. [Online]. Available: https://learn.adafruit.com/adafruit-si1145-breakout-board-uv-ir-visible-sensor/overview. [Accessed: 28- Mar- 2019].

[20]*Cdn-shop.adafruit.com*. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/Si1145-46-47.pdf. [Accessed: 28- Mar- 2019].

[21]A. Industries, "SI1145 Digital UV Index / IR / Visible Light Sensor", *Adafruit.com*. [Online]. Available: https://www.adafruit.com/product/1777. [Accessed: 28- Mar- 2019].

[22]S. Thing, "SparkFun ESP32 Thing - DEV-13907 - SparkFun Electronics", *Sparkfun.com*. [Online]. Available: https://www.sparkfun.com/products/13907. [Accessed: 28- Mar- 2019].

[23]E. Systems, "ESP32-WROOM-32 Espressif Systems | RF/IF and RFID | DigiKey", *Digikey.com*. [Online]. Available: https://www.digikey.com/product-detail/en/espressif-systems/ESP32-WROOM-32/1904-1010-1-ND/8544305. [Accessed: 28- Mar- 2019].

[24]"ESP32-WROOM-32", *Espressif.com*, 2019. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Accessed: 28- Mar- 2019].

[25]"Advantages and limitations of the Different Types of Batteries - Battery University", *Batteryuniversity.com*. [Online]. Available: https://batteryuniversity.com/learn/archive/whats_the_best_battery. [Accessed: 28- Mar- 2019].

[26]"Different Types of Batteries and their Applications", *Circuitdigest.com*. [Online]. Available: https://circuitdigest.com/article/different-types-of-batteries. [Accessed: 28- Mar- 2019].

[27]"Solar Charge Controller Basics | Northern Arizona Wind & Sun", *Solar-electric.com*. [Online]. Available: https://www.solar-electric.com/learning-center/batteries-and-charging/solar-charge-controller-basics.html. [Accessed: 28- Mar- 2019].

[28]"I2C - learn.sparkfun.com", *Learn.sparkfun.com*. [Online]. Available: https://learn.sparkfun.com/tutorials/i2c/all. [Accessed: 28- Mar- 2019].

[29]"BLE Advertising", *Argenox*. [Online]. Available: https://www.argenox.com/library/bluetooth-low-energy/ble-advertising-primer/#advertising-bluetooth-services. [Accessed: 28- Mar- 2019].

[30]"Bluetooth low energy Services, a beginner's tutorial", *Nordic DevZone*. [Online]. Available: https://devzone.nordicsemi.com/tutorials/b/bluetooth-low-energy/posts/ble-services-a-beginners-tutorial. [Accessed: 28- Mar- 2019].

[31]"GATT Services | Bluetooth Technology Website", *Bluetooth.com*. [Online]. Available: https://www.bluetooth.com/specifications/gatt/services. [Accessed: 28- Mar- 2019].

[32]"Bluetooth low energy Services, a beginner's tutorial", *Nordic DevZone*. [Online]. Available: https://devzone.nordicsemi.com/tutorials/b/bluetooth-low-energy/posts/ble-services-a-beginners-tutorial. [Accessed: 28- Mar- 2019].

[33]"Viewer | Bluetooth Technology Website", *Bluetooth.com*. [Online]. Available: https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.service.battery_service.xml. [Accessed: 28- Mar- 2019].

[34]"Viewer | Bluetooth Technology Website", *Bluetooth.com*. [Online]. Available: https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.service.environmental_sensing.xml. [Accessed: 28- Mar- 2019].

[35]"Web Services Architecture". World Wide Web Consortium. 11 February 2004. 3.1.3 Relationship to the World Wide Web and REST Architectures. Retrieved 29 September 2016.

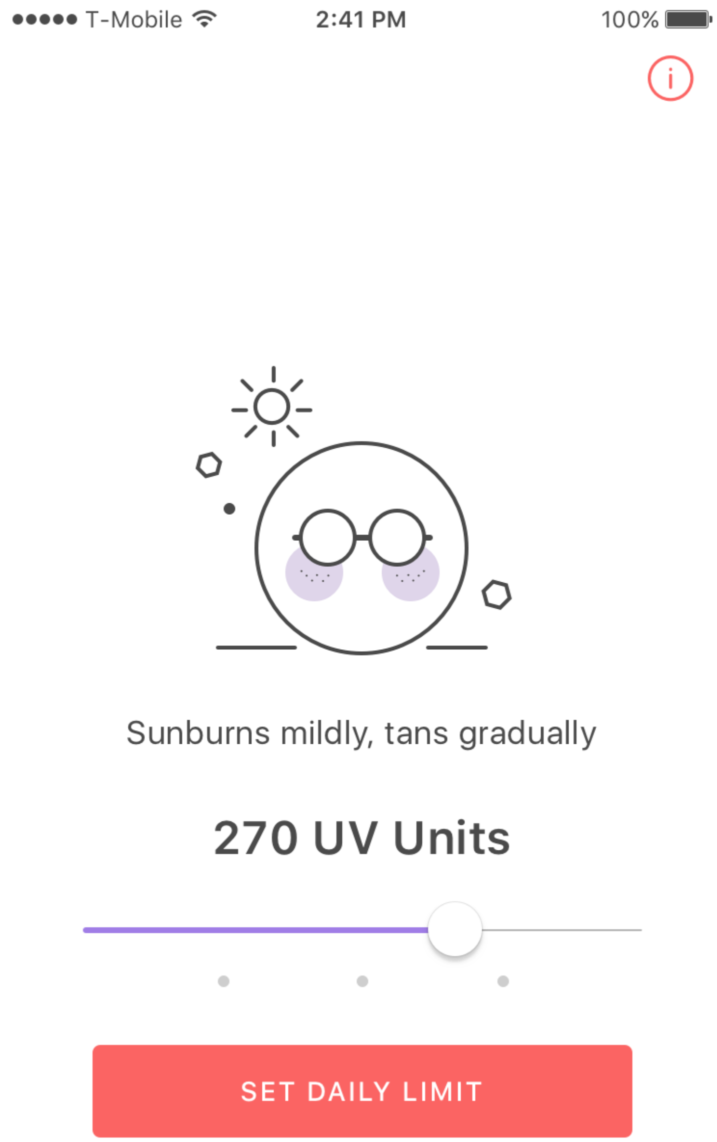
[36]"What is Scrum?", *Scrum.org*. [Online]. Available: https://www.scrum.org/resources/what-is-scrum. [Accessed: 28- Mar- 2019].

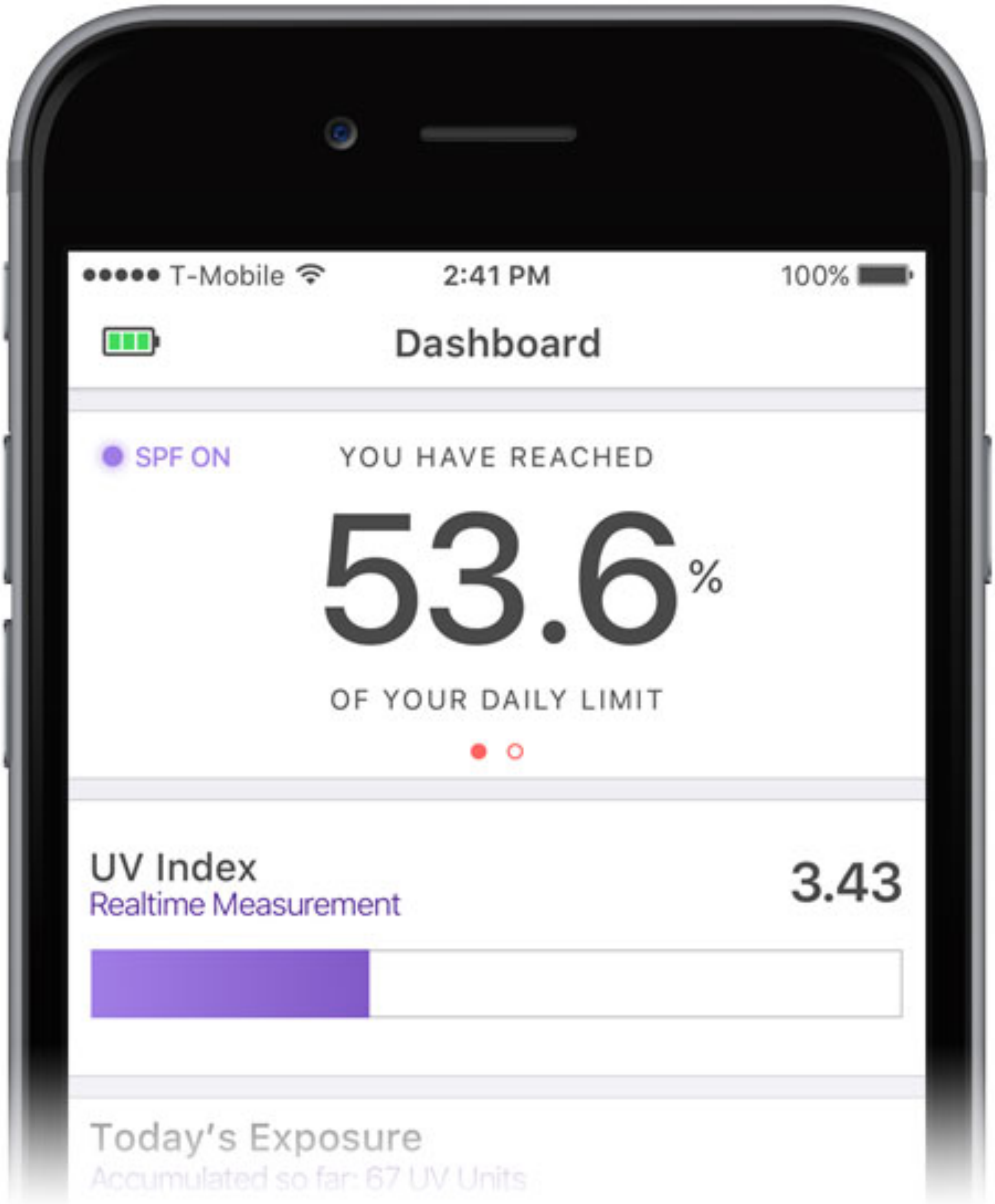
[37]"What is a Sprint Review?", *Scrum.org*. [Online]. Available: https://www.scrum.org/resources/what-is-a-sprint-review. [Accessed: 28- Mar- 2019].

[38]"What is a Sprint Retrospective?", *Scrum.org*. [Online]. Available: https://www.scrum.org/resources/what-is-a-sprint-retrospective. [Accessed: 28- Mar- 2019].

Appendices

Appendix A — Screenshots from Shade's iOS application.

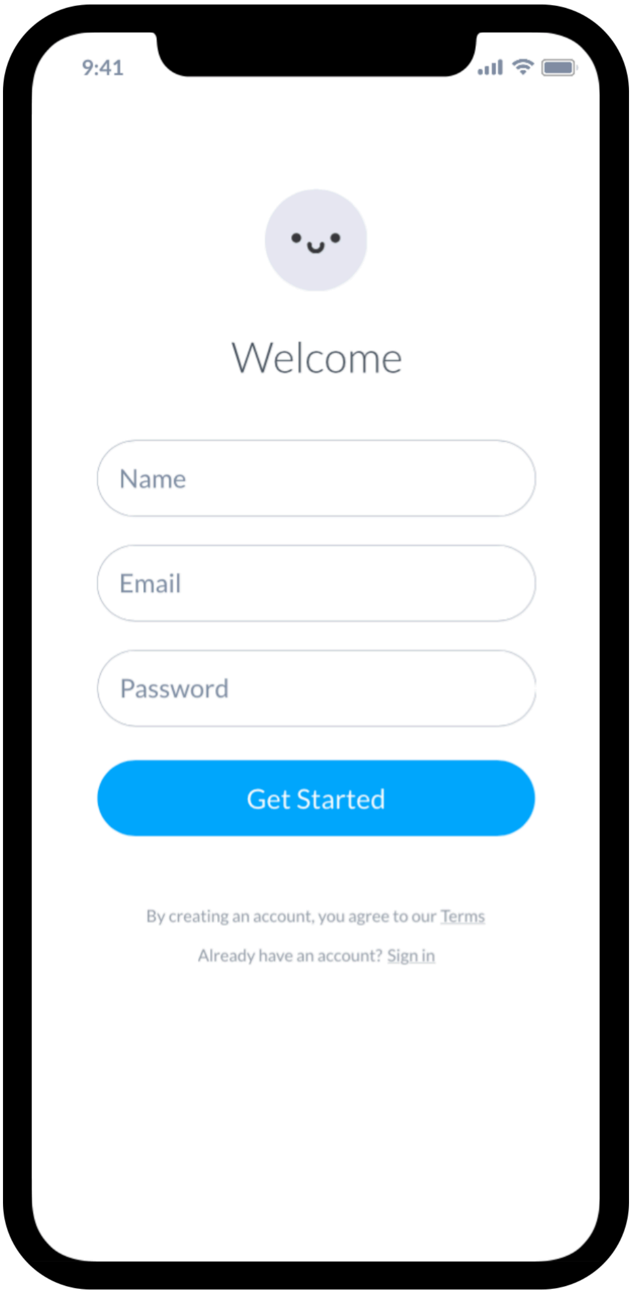


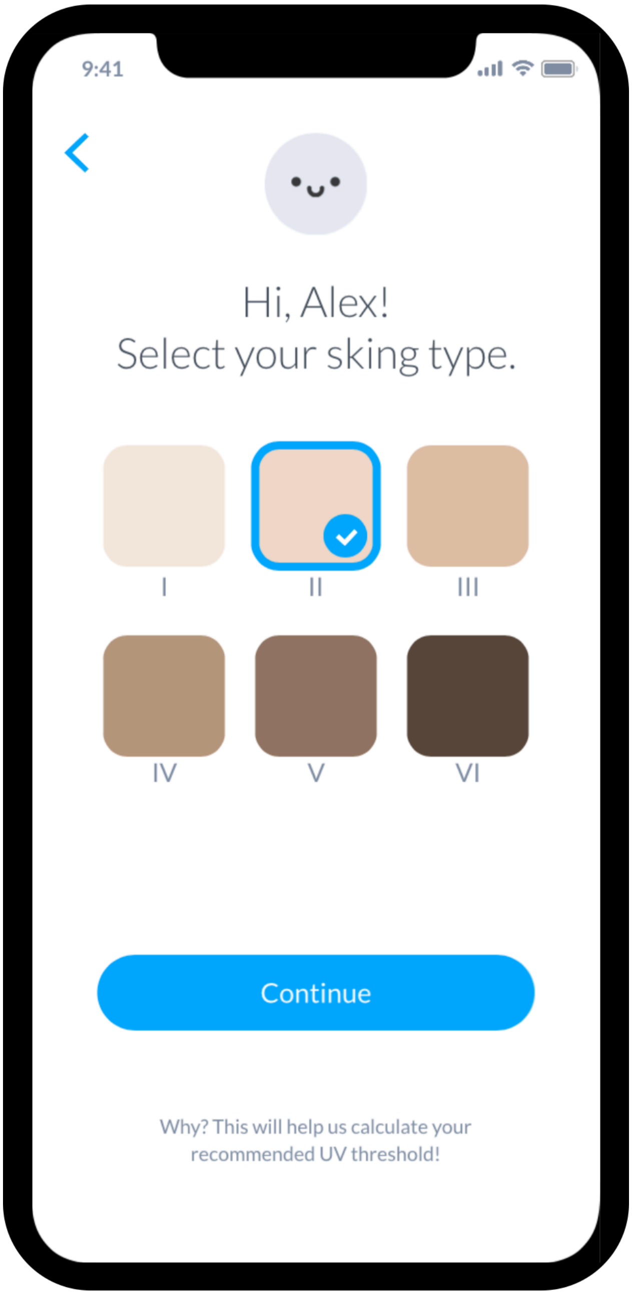

T-Mobile
2:41 PM
100%
Dashboard
SPF ON
YOU HAVE REACHED
53.6%
OF YOUR DAILY LIMIT
UV Index
Realtime Measurement
3.43
Today's Exposure
Accumulated so far: 67 UV Units

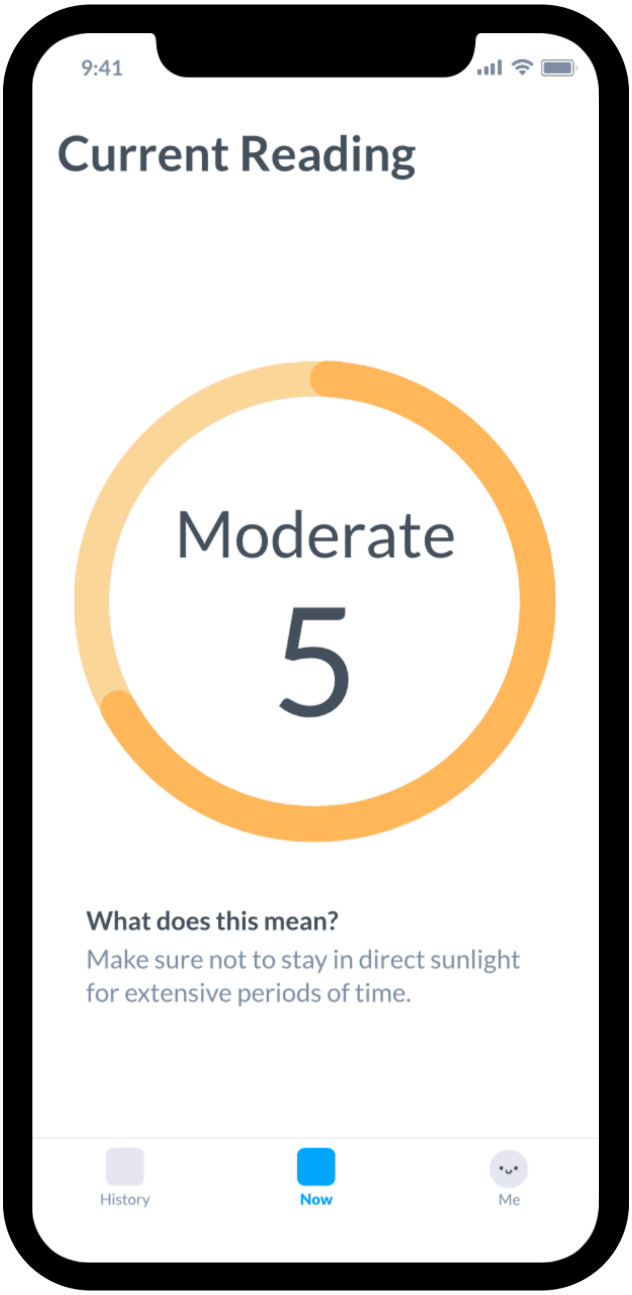
Appendix B — Screenshots from QSun's iOS application

i.e. trello board for scrum planning

Appendix C — Design prototype for smartphone application.




9:41
Hi, Alex!
Select your sking type.
I
II
III
IV
V
VI
Continue
Why? This will help us calculate your recommended UV threshold!


9:41
Current Reading
Moderate
5
What does this mean?
Make sure not to stay in direct sunlight for extensive periods of time.
History
Now
Me


9:41
History
This Week
nday
Tuesday
Wednesday
Thursday
Fri
Exposure
10 hr 18 min
Moderate
5 hr 18 min
High
3 hr
Very High
2 hr
This Month
History
Now
Me