PocketClimate:

Harnessing Smartphone Sensors for Personalized Climate Awareness

CS7470 - Mobile & Ubiquitous Computing

Pocket Climate - Proposal

Team 31

Valentina Calabrese
Georgia Institute of Technology
vcalabrese3@gatech.edu

Cheyenne Ralston
Georgia Institute of Technology
cralston8@gatech.edu

Omari Maxwell
Georgia Institute of Technology
omaxwell3@gatech.edu

Mayowa Adegeye
Georgia Institute of Technology
madegeye3@gatech.edu

ABSTRACT

The growing ubiquity of mobile technology has provided an opportunity to leverage embedded sensors in smartphones to provide a myriad of services to users. In this context, our project proposes to design and implement a mobile application that utilizes a smartphone's inbuilt sensors to provide real-time microclimate monitoring and an indoor comfort index. The motivation behind this project lies in the need for personal, localized environmental awareness, aiming to help users understand their immediate surroundings better and make informed decisions about their indoor comfort. We aim to create an app that effectively gathers and processes sensor data, presenting it in a user-friendly and meaningful way.

AIMS & OBJECTIVES

Our primary aim is to develop "PocketClimate", a mobile app designed to tap into the potential of built-in smartphone sensors for the purpose of offering personalized climate awareness.

Understanding Smartphone Sensors: Our first objective involves conducting an in-depth analysis of the capabilities and limitations of typical smartphone sensors. Our focus will be on sensors that can provide climate-related data such as the barometer, light sensor, and potentially a hygrometer.

Data Gathering and Processing: We'll need to develop a reliable back-end system that can gather data from these sensors and process it accurately. This involves interpreting raw sensor data into meaningful weather-related parameters and creating a comfort index based on these parameters.

User-friendly Interface: We'll be designing and developing a front-end interface that is intuitive and visually appealing. It should effectively present the sensor data and comfort index to the user, allowing them to easily understand their current microclimate.

Comprehensive Quality Assurance: Another critical objective is the thorough examination of the app's functionality and user experience across various devices and conditions. This involves a comprehensive alpha and beta testing strategy, allowing us to identify and remedy any flaws or performance issues. The insights gained during these stages will enable us to refine the app, ensuring a seamless and intuitive user experience.

BACKGROUND

There is existing work in the smart weather detection space pertaining to embedded electronics with sensor technology [1], [2], [3] use of environmental data to make forecasts [4]. Research has covered the use of cloud technology to power the backend of a weather technology platform as a method of resource optimization in a ubicomp solution. Faid et al provided a solution that "relies on a fully containerized architecture, where services are not only lightweight but scalable, agile, and portable. This makes the … full software architecture manageable and reproducible within a very short amount of time. It allows us to have very reliable and optimized micro-services such as local web server, time-series databases" [5]. Another area of exploration is in keeping the exorbitant costs of building a smart-home down and the benefit of widely available weather station smart systems in developing economies. Sima et al state "it is necessary to support families in the decision-making process, facilitating the use of smart systems through the IoT, wireless technologies, and with low-cost open source hardware and software" [6].

CHALLENGES AND ALIGNMENT TO CLASS

Some of the challenges we need to overcome to deliver this project include the following:

Accuracy of sensors: There are a litany of mobile devices out there each from different manufacturers and have different capabilities. Our project is highly dependent on getting accurate data from these sensors which might function differently depending on the manufacturer. Some sensors are also known to work better under certain environmental conditions which might not be readily available.

Battery life of the mobile device: Since our application is going to run on a mobile phone which is resource constrained, we have

to ensure that we read data from the sensors in such a way that optimizes the mobile phone's resources. Sensors such as gyroscopes and GPS are known to consume a lot of device power and we need to factor this into our implementation.

Synchronization of readings from multiple devices: Since our application involves getting data from multiple sensors on the device such as a barometer, gyroscope, accelerometer, etc, there is no need to synchronize the readings from these sensors so as to be able to sense them to the backend as a batch and make a useful inference from it.

Scalability of the backend service: We need to develop a backend service scalable enough to handle a stream of sensor data coming from the mobile application.

Choice of mobile OS platform: Since our application is going to be deployed on mobile devices, one of the challenges we need to solve is the choice of mobile OS platform. We are faced with the challenge of building a native solution for each of the popular platforms or going for a cross-platform technology that bridges this gap. A major challenge with cross-platform technologies is that some of them don't integrate properly with some of the device hardware that our application depends on

Ubiquitous computing involves adding computing capabilities and context awareness into everyday objects in such a way that makes them pervasive and able to report data that are context sensitive.

Our proposed application aligns perfectly with this goal by making use of context-aware components on a ubiquitous computing device like the mobile phone to report environmental data like temperature, humidity, etc. It is always on and requires little to no interaction from the user to perform its task. This makes it align with what we have learned so far in the class.

OUTCOMES & DELIVERABLES

Our targeted deliverables will include:

1. Final UX Mockups:
    1.1. High fidelity mockups of our intended application screens including basic use case flows
2. Software Application:
    2.1. Android application that makes use of built-in smartphone sensors to harvest weather related data
    2.2. Data model showing what data is being collected and what is being done with the data at each phase of use
    2.3. Scalable backend to handle the various data collected from sensors
3. Data from Testing:
    3.1. Results from testing our user scenarios in the final application
    3.2. Test plan detailing what tests were done to evaluate the quality and reliability of the user experience

PROJECT PLAN & TIMELINE

We have divided our project up into three distinct phases: Design and Planning, Physical Prototyping and Development, and Testing. Each of these three phases will carry out tasks related to analysis, refinement, testing, and evaluation as necessary with some flexibility to adjust as the project progresses. We expect that there will be some overlap between these phases, and will aim to create an iterative process that allows us to respond to any obstacles or blockers in a timely manner.

Phase 1 Design and Planning: This phase will involve us fully defining the problem space we will be working in. To that end, we will conduct an analysis to define the user requirements, which will allow us to understand how best to address the needs of our users. Once these have been defined, we will design an architectural solution that suits our use case, including what mobile sensors will be used in the app. The last portion of this phase will be centered around creating wireframes and mockups of the mobile app. This will help us get a better feeling for the flow and navigation of the app. We aim for these three milestones to be completed by the end of the first week, although there is some possibility of them spilling over into the second week.

Phase 2 Physical Prototyping and Development: This phase will center around actually developing the mobile app and making sure the sensors are feeding data into it appropriately. There are three separate data points we are attempting to measure and display: temperature, pressure, and humidity. Each of these has a task to write code to retrieve the sensor data, and then create front end logic to visualize the data within the mobile app. This phase will likely be the most burdensome of the three phases, and we anticipate that we will spend the bulk of our time in this phase and thus have given it the most stretch time of approximately four weeks, leaving almost two weeks for testing and refinement.

Phase 3 Testing: This phase is the one in which we will conduct testing on our mobile app, making sure it suits the initial user requirements. Although unit tests were likely developed alongside the code, we will create an end-to-end testing scenario for the entire mobile app and carry it out to identify any bugs we may have missed. This phase will also include the creation of the reports and videos necessary for the final project. This phase is planned for approximately the last ten days, though there will likely be some overlap with the previous phase.

REFERENCES

[1] Gao, S. (2022). Research on smart home system based on photoresistor PIR sensor. In Proceedings of the Second International Conference on Cloud Computing and Mechatronic Engineering (I3CME 2022) (Vol. 12339, pp. 500-505). SPIE. DOI:https://doi.org/10.1117/12.2654949

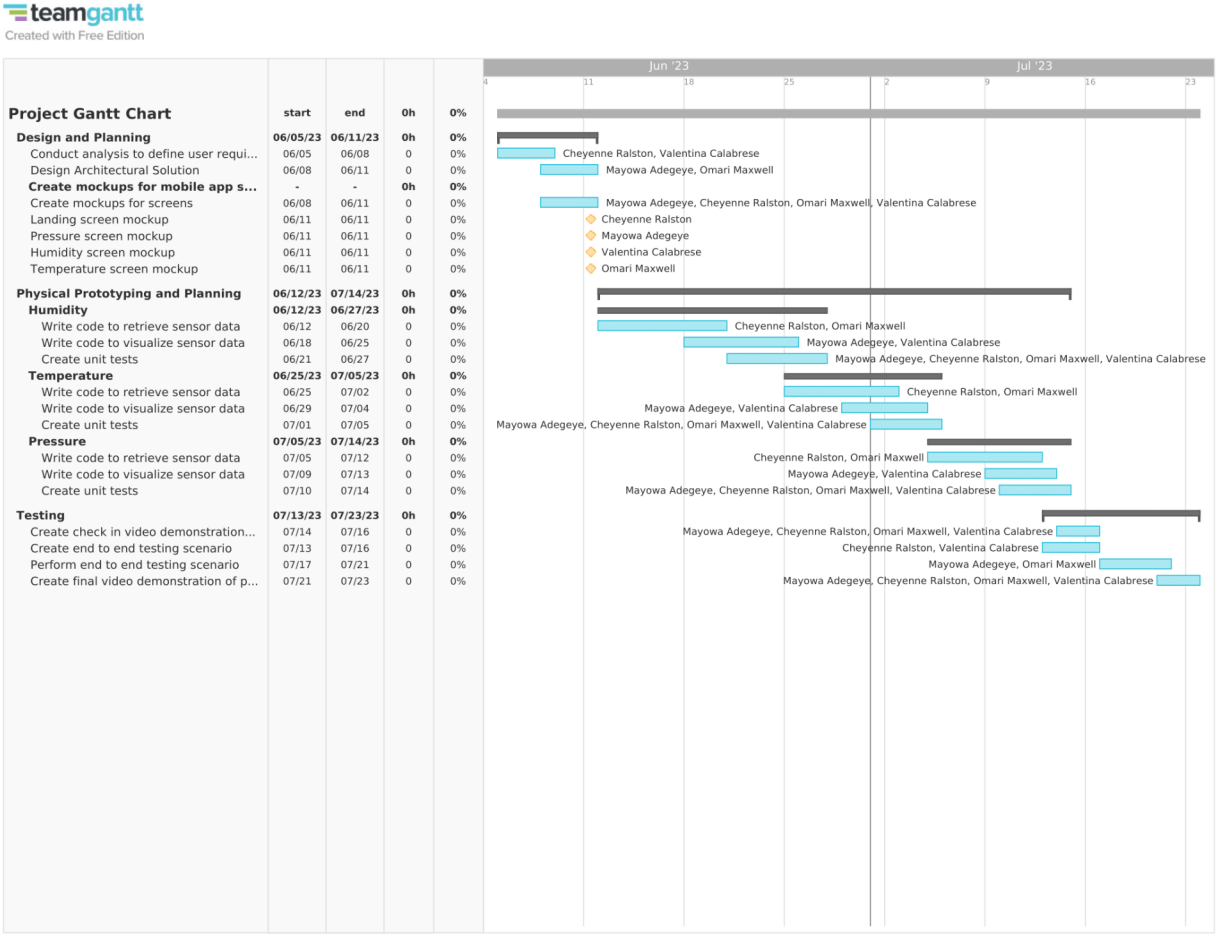
[2] Molnár, J., Kiresova, S., Vince, T., Kovac, D., Jacko, P., Beres, M., & Hrabovský, P. (2020). Weather Station IoT Educational Model Using Cloud Services. Journal of Universal Computer Science, 26(11), 1495-1512.

[3] Faid, A., Sadik, M., & Sabir, E. (2022). An agile AI and IoT-augmented smart farming: a cost-effective cognitive weather station. Agriculture, 12(1), 35. DOI:https://doi.org/10.3390/agriculture12010035

[4]Djordjevic, M., & Dankovic, D. (2019). A smart weather station based on sensor technology. Facta Universitatis, Series: Electronics and Energetics, 32(2), 195-210. DOI:https://doi.org/10.2298/FUEE1902195D

[5]Hussein, Z. K., Hadi, H. J., Abdul-Mutaleb, M. R., & Mezaal, Y. S. (2020). Low cost smart weather station using Arduino and ZigBee. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(1), 282-288. DOI:https://doi.org/10.12928/telkomnika.v18i1.12784

[6]Sima, C. A., Popescu, C. L., Popescu, M. O., Grigorescu, S. D., & Roscia, M. (2022, June). Weather Station Smart Home System for Sustainable Cities in Developing Economies. In 2022 IEEE International Conference on Environment and Electrical Engineering and 2022 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe) (pp. 1-6). IEEE. DOI:https://doi.org/10.1109/EEEIC/ICPSEurope54979.2022.9854632



Project Gantt Chart	start	end	0h	0%
Design and Planning	**06/05/23**	**06/11/23**	**0h**	**0%**
Conduct analysis to define user requi...	06/05	06/08	0	0%
Design Architectural Solution	06/08	06/11	0	0%
Create mockups for mobile app s...	**-**	**-**	**0h**	**0%**
Create mockups for screens	06/08	06/11	0	0%
Landing screen mockup	06/11	06/11	0	0%
Pressure screen mockup	06/11	06/11	0	0%
Humidity screen mockup	06/11	06/11	0	0%
Temperature screen mockup	06/11	06/11	0	0%
Physical Prototyping and Planning	**06/12/23**	**07/14/23**	**0h**	**0%**
Humidity	**06/12/23**	**06/27/23**	**0h**	**0%**
Write code to retrieve sensor data	06/12	06/20	0	0%
Write code to visualize sensor data	06/18	06/25	0	0%
Create unit tests	06/21	06/27	0	0%
Temperature	**06/25/23**	**07/05/23**	**0h**	**0%**
Write code to retrieve sensor data	06/25	07/02	0	0%
Write code to visualize sensor data	06/29	07/04	0	0%
Create unit tests	07/01	07/05	0	0%
Pressure	**07/05/23**	**07/14/23**	**0h**	**0%**
Write code to retrieve sensor data	07/05	07/12	0	0%
Write code to visualize sensor data	07/09	07/13	0	0%
Create unit tests	07/10	07/14	0	0%
Testing	**07/13/23**	**07/23/23**	**0h**	**0%**
Create check in video demonstration...	07/14	07/16	0	0%
Create end to end testing scenario	07/13	07/16	0	0%
Perform end to end testing scenario	07/17	07/21	0	0%
Create final video demonstration of p...	07/21	07/23	0	0%

Link to pdf version of the gantt chart: https://drive.google.com/file/d/13xeagAxtgLBArvAAav7ghtwLRjgmQ5qq/view?usp=sharing