CS 4337 Human-Computer Interaction

Help-Me IT

Authors:

* Andrew Samuel
* Jacquelyn Johnson
* Paul Davis
* Reggie Andes

Submission Date: August 9, 2020

# Contribution Breakdown

|  |  |
| --- | --- |
| **Member** | **Contributions** |
| Andrew Samuel | 1. Co-designer 2. Wrote all initial dialog for app Q&A 3. In-charge of assuring project report completion 4. Created layout for project report 5. Handled evaluation strategy 6. Compiled project’s conclusion statement 7. Contributed to presentation content |
| Jacquelyn Johnson | 1. Co-designer 2. Defined the app concept 3. Initialized repository 4. Held lesson to improve team understanding of git 5. Created and implemented app icon 6. Created templates for common activities 7. Implemented settings menu 8. Customized dialog for mobile category 9. Implemented Mobile Category 10. Created sliding transitions between activities 11. Planned and collaborated on video demo 12. Produced Scenarios |
| Paul Davis (Team Leader) | 1. Co-designer 2. Determined project schedule 3. Created SPMP 4. Designed Crowdsource prototype 5. Created empty activities and appropriate labeling for each category. 6. Wrote code for radio button logic and opening new activities 7. Customized dialog for laptop/desktop category 8. Implemented Laptop/Desktop Category 9. Initiated and outlined test case creation 10. Created presentation outline and content 11. Created and edited video demo 12. Produced Personas |
| Regina Andes | 1. Co-designer 2. Gathered image resources 3. Implemented Home screen 4. Customized dialog for console category 5. Implemented Console Category 6. Implemented color schemes 7. Test case creation 8. Contributed to presentation content |

# 

# Motivation

Help-Me IT is an entertaining app which is meant to be a parody of painful to use technical support apps. The motivation for this project came from the British sitcom *The IT Crowd,* which features characteristically lazy IT employees. The app offers a small amount of actual troubleshooting followed quickly by sarcastic or judgemental remarks. Potential users include anyone that is looking to pull a quick prank on a friend, people that are tired of offering free technical support, or individuals that are looking for tech-support humor. The benefit we hope to bring is a fun app full of sarcastic and jovial comments to give the user a good laugh.

# Software Development Life-Cycle

We chose the waterfall model to develop our app. Many of us are working with Android Studio for the first time, so skill limitations are important to recognise. We aimed to keep things simple and easy to manage. Because the product requirements are documented and approved prior to the beginning of development, we can assure the delivery of a specific set of features which makes the final product more predictable.

# Personas, Scenarios, and Use Cases

# Personas

|  |  |
| --- | --- |
| **Background**   * Tech Guru * 30 year-old male * Lives alone * Leases apartment | **Dave**  Computer Generated Image  Dave’s friends and family always come to him for free technical support. Dave usually makes excuses to get out of it. When he is sick of repeating himself, he knowingly refers people to alternative dead-end resources. |
| **Motivations**   * Advancing his career * New technology * Knowing everything * Fixing everything on his own * Being recognized as superior |
| **Frustrations**   * Incompetence * Poor understanding of technology * Dumb questions * Wasting time |

Dave could benefit from Help-Me IT by having the option to refer inquiries to the app. Instead of having to actually do any work, he may inform the customer/coworker to fix it themselves by going through the troubleshooting questions found in Help-Me IT for the category they are needing assistance with.

|  |  |
| --- | --- |
| **Background**   * Seamstress * Renaissance Festival employee * 55 year-old female * Lives with several cats | **Karen**  Computer Generated Image  Karen prefers to ask for help before trying anything herself. She will avoid learning new skills at all costs. |
| **Motivations**   * Making macrame owls * Speaking to managers * Watching her neighbors * Making orders on the shopping network. |
| **Frustrations**   * Computers * Unclear instructions * Being asked to wear a mask * Waiting in line * Being expected to keep her receipt |

Karen is likely to misunderstand Help-Me IT’s purpose. The instructions are clear and easy to follow, so she is likely to understand the layout of the app. Karen represents the audience that is most likely to be referred to Help-Me IT by tech-savvy individuals as a joke.

# Scenarios

* + 1. Jacque is a tech guru who frequently is asked to fix hardware issues that arise on machines that belong to other people. These issues are usually easy fixes and she didn’t mind the first time that she was asked to fix an old laptop for an elderly person who found out that she had a degree involving computers. However, Jacque’s degree has nothing to do with hardware. Ever since she helped the first person, news has spread at her church that she can fix computer problems. Jacque longs for a respite from the incessant requests to do this work for free as this is taking time away from making improvements on a piece of software that she is developing. Jacque finds the HelpMe-IT app on the Android store and is thrilled to direct others to it so they can do their own troubleshooting and hopefully be entertained in the process.

# Use Cases

|  |  |
| --- | --- |
| **User Intention** | **System Responsibility** |
| Open app | Display main screen |
| Select category | Open correct activity based on image button selected |
| Select radio button and clicks submit | Display text and relevant button to continue troubleshooting |
| Selects ‘YES’ | ‘NO’ during troubleshooting | Display text and ‘NEXT’ | ‘RETURN TO MENU’ button based on selection. |



# Prototypes

To begin our design phase, we explored the existing markets for related apps so that we may evaluate design and ease-of-use. This yielded two primary apps that would inspire our user interface (UI). Each design was examined by our team members and judged by subjective satisfaction and feasibility. First and foremost, it was required that the selected design support the conversational interaction that the team had envisioned. Secondly, the interface must be simple to use and instructions must be clear. The first app we evaluated was Xfinity Assistant (Figure 1), which features a chat-based design that allows the user to provide open-ended responses. For Help-Me IT, each user response must have a corresponding pre-planned retort. Because of this, the chat-design was nixed so we could incorporate a more controlled back-and-forth between the application and the user to ensure that our responses would make the most sense in context.

A screenshot of a cell phone

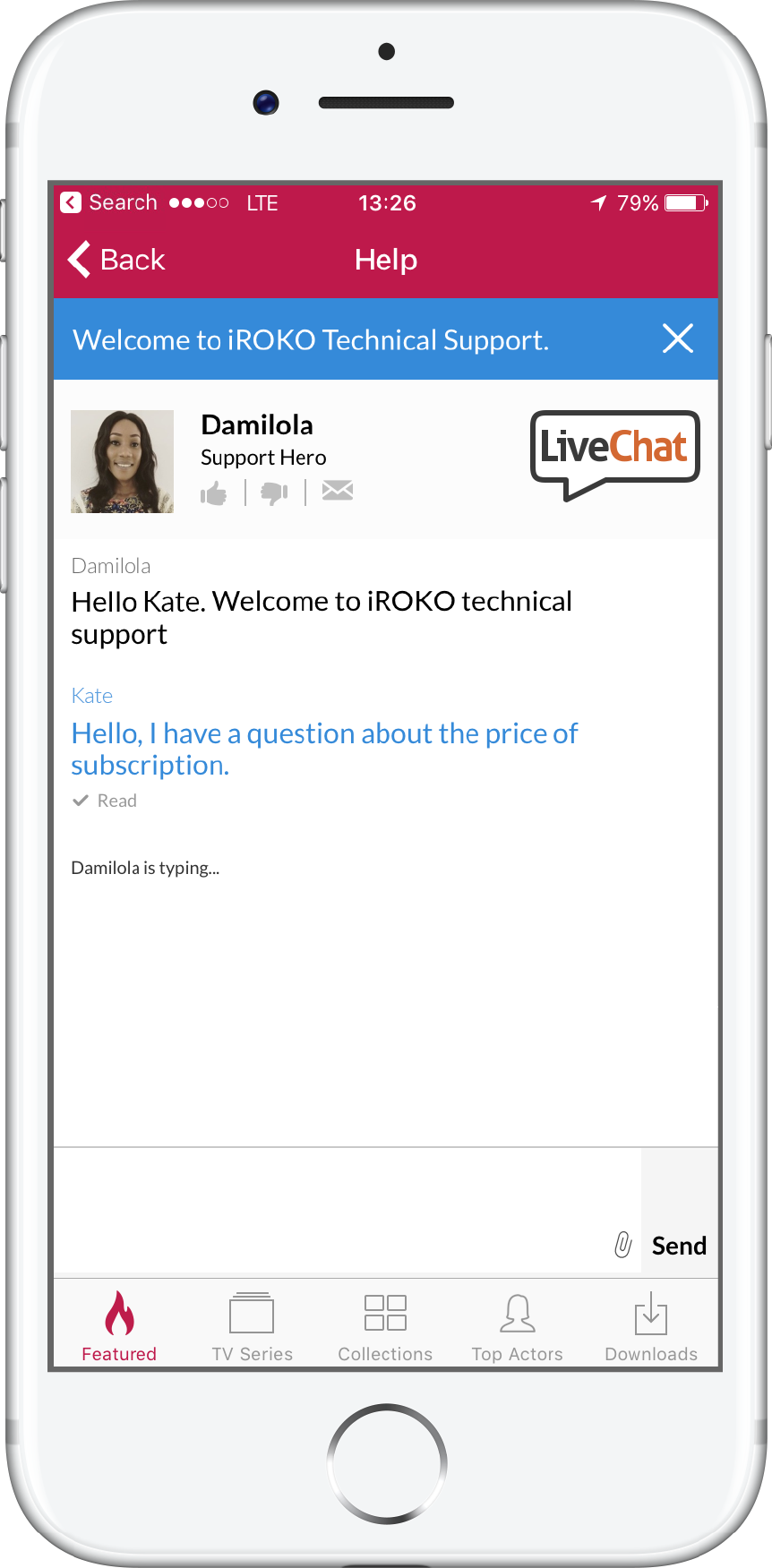
Description automatically generated

Fig 1. Xfinity Assistant and similar chat-based app interface

The next design idea was formed around controlled question and answer prompts. This led to us a Google Crowdsource (Figure 1) inspired design for our prototype. The prototype home screen contained four image buttons depicting categories for technical support. Upon clicking one of the image buttons, the user is prompted with inquiries about their problem. As those questions are answered, the sarcastic quips appear on the screen. The prototype was evaluated and decided, by the team, as our basis for design.

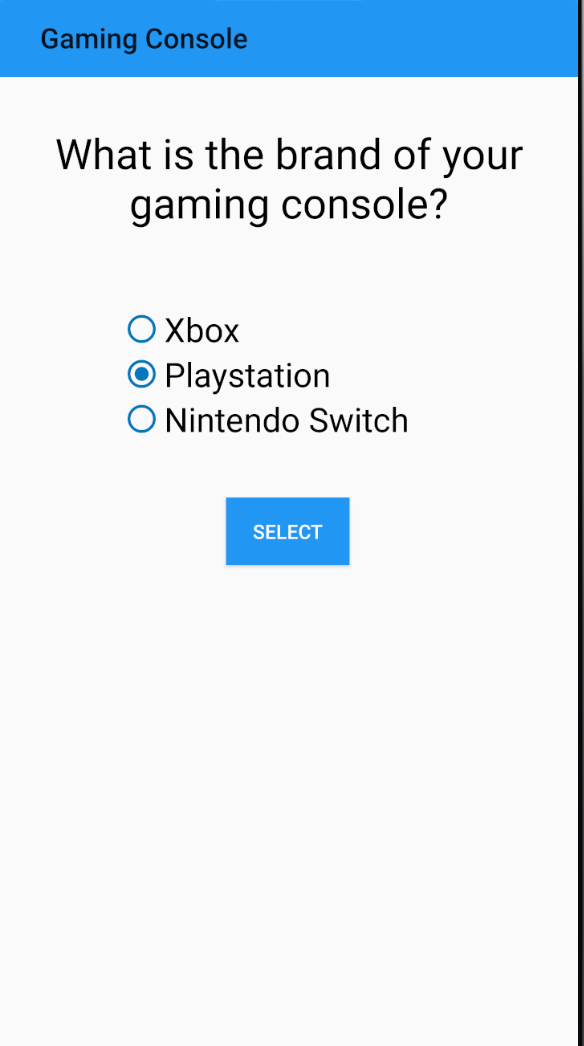
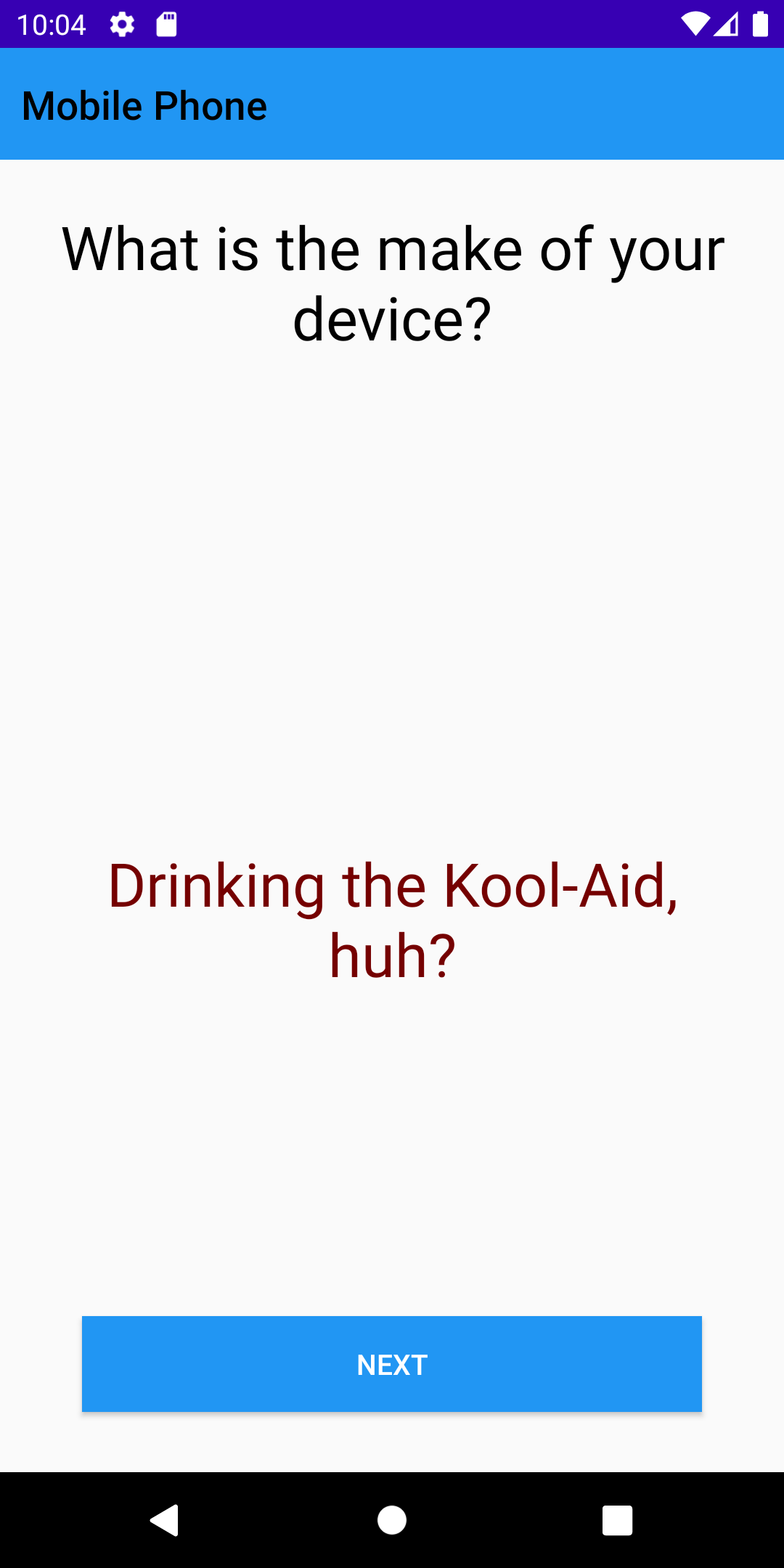
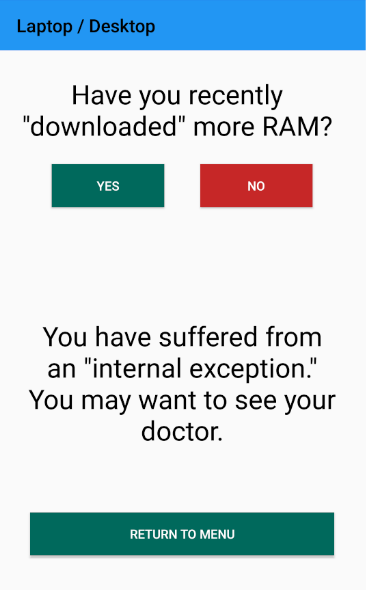
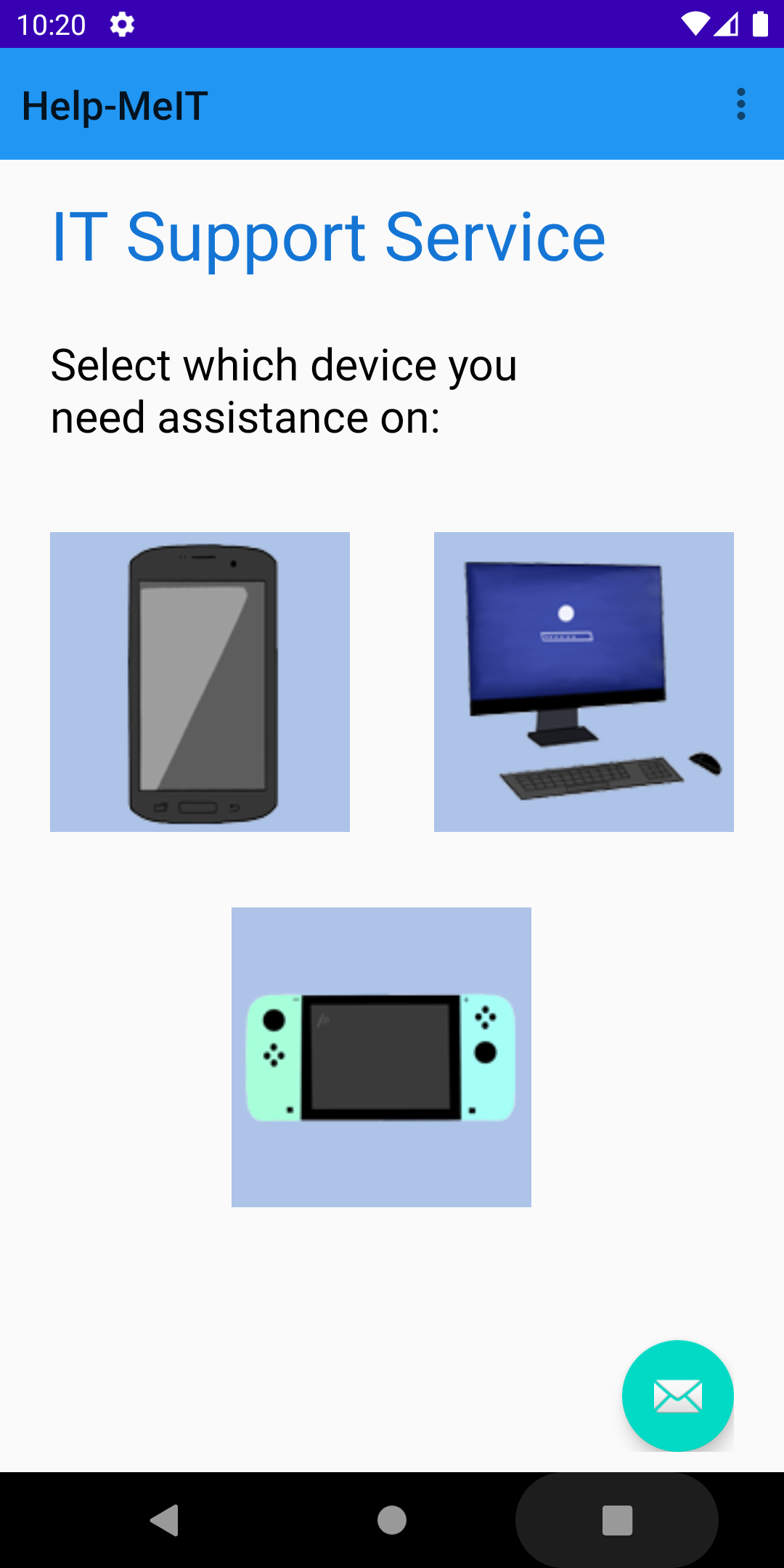
A screenshot of a cell phone

Description automatically generated

Fig 2. Google Crowdsource inspired prototype

# User Interface

The home screen of the final product features three image buttons that represent categories of support that the user may select. Upon making a selection, a slide transition brings the user to a new activity that probes the user with questions about their device. The first line of questioning determines the device make/model. They are given preset responses presented on either clickable buttons or radio groups. We have themed our buttons using familiar green/red color schemes for “Yes”/”No” replies. The theme and color scheme throughout the app follows consistently. Based on the user’s response to the questions, the app displays sarcastic or humorous commentary while prompting the user to hit “Next” or “Return to Home Screen” depending on the context.

****

# Evaluation Plan

After conducting formative evaluation procedures throughout the development of the app, we began the next phase of our evaluation process once the majority of the app functions were finalized. We began with our programmers doing small usability tests as summative and diagnostic evaluations on their assigned sections. We did this by running an emulator of the Pixel on Android Studio as an AVD (Android Virtual Device). In addition to that we ran a field study involving peoples outside our group running the app on their personal android phones to make sure no issues were found when translating the app from an AVD to an actual android device. This also proved useful for finding logic problems and bugs that we might have overlooked or overstated in our findings. Our last step in the evaluation phase was the measurement evaluation. As part of the measurement evaluation we ran the app in Android Studio using the Pixel AVD emulator and ran an app profile noting down the install time, app restart time, the time it took to transition between questions, and the total memory used. The app took 1 second and 173 milliseconds to successfully install and run for the very first time and 106 milliseconds to successfully restart without requiring a reinstallation. For the transition of activities to new questions the app averaged approximately 2 milliseconds per transition to get to the next question of the category. Lastly, the total memory used accumulated to 96 MB. We intend on continuing evaluation protocol as this app is pushed onto a consumer field study and into the market space to maintain the function of the app.

# Conclusion

The creation of Help-Me IT presented our team with the daunting challenge of pure virtual collaboration. However, each obstacle was ultimately surmounted. We took advantage of a number of technologies that would aid with virtual collaboration, including GitHub, Google Docs, and Zoom. Schedule conflicts and an unreliable power grid forced our team to be dynamic in the scheduling of meetings and tasks. The project initially adopted a democratic team dynamic, however, it became clear that a team leader would improve the division of work. Once the conceptual design and the user interface was decided upon, our next steps were to designate the leader and divide up tasks. After tasks were assigned, we put our entire project on GitHub to establish version control and a communal space to place our project materials. This allowed us enhanced organization; keeping track of the project as it went on in one designated place. We faced hundreds of merge conflicts on GitHub initially, but this problem was solved when it was discovered that the “.gitignore” file had not been generated. During the course of implementation, it became clear that the questioning and retorts were far too similar in each category and lacked uniqueness. This problem was resolved as we broke down the template that we had and began to go question by question to make the device categories and the overall app unique. Lastly, we prepared a project presentation which presented the project motivation, functions within the app, and our evaluation plan for our project. However, we faced an additional challenge when our lead programmers could not participate in the live presentation due to scheduling conflicts. We worked around that issue by recording the presentation at an earlier date. As we move forward, our team will continue to maintain the app and add additional features, such as an opportunity for the user to rate our service. We intend to eventually attain a quality that would be worthy of addition to the Android marketplace so that the world may enjoy using it as much as we have enjoyed creating it.

**Project-Plan Scenario**

|  |  |
| --- | --- |
| **Timeline** | **Tasks** |
| Week 1 (7/6-7/12) | Requirements / Design   1. Requirements Analysis 2. Use-case analysis 3. Select a user-interface theme |
| Week 2 (7/13-7/19) | Design / Implementation   1. Finalize support categories, specific support issues, and technical support responses. 2. Select or create images for buttons. 3. Develop each screen layout |
| Week 3 (7/20-7/26) | Implementation / Integration   1. Fill in each screen layout with information from the design phase. 2. Create transitions between screens. |
| Week 4 (7/27-8/2) | Integration/Testing   1. Determine test cases 2. Perform test cases 3. Debug |
| Week 5 (8/3-8/9) | Evaluation   1. Conduct peer reviews. 2. Create product summary poster |

# References

**Websites used:**

Android Studio documentation---<https://developer.android.com/>

draw.io---<https://www.draw.io/>

GitHub---<https://github.com/>

Google Drive---<https://drive.google.com/>