Developers: Crystal Gong (cjg5uw), Annie Vu (dv2ku)
Aipom/Android

Self-Care-Bear

Description

Self-Care-Bear is a task keeper geared towards self-care. The tasks are simple, and upon completion of a task, you are greeted with words of positive encouragement and accumulation of points, gamifying the completion of tasks. Self-Care Bear (the panda) wants you to want yourself to succeed and thrive. In return for her encouragement, you can buy Self-Care Bear gifts from the Gift Shop with the points that you earn so that she also has a more enriched life.

Platform Justification

Android is more familiar to the both of us since it uses java as opposed to swift. Additionally, because there exists more sample code on the internet for building Android apps than there does for Swift 3, we reasoned that it would be easier and less frustrating to learn on Android. Android, without drag and drop tools like XCode has, forced us to learn how to code things programmatically, which gave us a more solid understanding what the code was actually doing than if we were to have used Swift. In terms of appearance, since our app has a background image and other images, the bigger screen of the tablet compared to the iPod gave us more space to add elements to the screen in a way that didn't feel cluttered. Lastly, because only one of two of us have a mac, working on Android was more convenient.

Key Features (include short descriptions of each)

- Main Screen
    - Panda Mama is in her house. She has various dialogues, reminding you to take care of yourself and encouraging you to do well. Tapping on her will rotate dialogues. Some dialogues depend on the time of day (morning, afternoon, or evening).
    - There's a task list that displays a list of tasks depending on the time of day. Checking off an item will add that item's point value to the user's overall point storage, which serves as an incentive to complete tasks.
- Navigation Menu
    - Use this menu to navigate from the Main Screen to the three other screens. Use the back button on android to return from any of these screens back to the Main Screen.
- Gift Shop
    - Buy gifts in exchange for Panda Points. Once a gift is purchased, the gift is displayed on the home screen for Panda Mama to enjoy.
    - One of the gifts brings up the **camera app** and lets you take a selfie (or any photo), which is then displayed on the wall.
- Task Manager
    - Here, you can add tasks to each of the three time periods.
    - Each task has a name, a time of day to which it's assigned, a number of Panda Points, and an optional location, which can be chosen with **Google's Place API**.
    - **If the location services is turned on in the settings and you choose a place that is farther than 2000m (about 1.25 miles) from your current location, then that task is worth twice the number of points as a reward for making the effort to travel a greater distance**

    - You can also delete tasks here.
- Google Calendar: In settings, you can turn on the option to sync your Google Calendar with your tasks.
    - **Pulling events into tasks**: It will sync with all the events currently visible on your google calendar UI for the afternoon or following day. The calendar will sync every 15 minutes.
    - **Turning tasks into events**: When Calendar is enabled, you can post any task that you have as an Calendar Event corresponding to the time of day and location (if applicable) of the task. The events are 15 minutes long and randomly distributed throughout the corresponding time period.
- Database
    - The app has its own personal database that is able to store all task items. A database helper class was made, and can store tasks, as well as read the tasks and delete and update them. While the database helper hasn't been made to properly store info in between app sessions, the class itself works perfectly fine (it has been tested extensively) and need only be implemented

Testing Methodologies (what did you do to test the app)

- Testing Calendar API: We ran the calendar event grabber continuously, making sure new tasks got added and would appear if I added them in calendar even after the timer started. Similarly, with posting events, I grabbed tasks from the three different categories and made sure they inserted into the correct time of day. Morning goes from 5am-9am, afternoon is from 9am to 6pm, evening is 6pm to 5am.
    - Wanted to prevent circular event/task creation, so tasks pulled from calendar cannot be made back into events, and events posted from the task list cannot become tasks again.
- Testing add task: We mostly just had to test the edge cases where some of the fields were null, and then implement a safeguard against the nulls. Added many many tasks, removed all of them.
- We rotated every single screen in every single state that we could imagine.
- I manipulated a bunch of data in all three of the fragment screens and made sure they matched up with the main screen list. Tried rotate every single screen on every single state after every combination of things. For add task, tried to input null for all fields.
- At first, to test our app we focused on the basic functionality. Specifically, making sure that all of our recyclerViews properly appeared and displayed our information in each fragment and activity. After finishing a small part, we would run the app and test to see if it's functional. If anything didn't work or created an error, we would know that something went wrong. Since there are many different ways to code a recyclerView, it was a bit of a challenge trying to build three different recyclerViews, each containing different information on a different screen. Since we split up the work, the way the recyclerViews were made are a bit different from each other- this made it a bit hard to debug each others' code, as the format is unfamiliar.
- Once basic frameworks were in place, we began focusing more on user experience and making the app easy and entertaining to use. This would be placing the buttons in

effective places, or arranging the different views in each activity a certain way so that it doesn't seem too cluttered. Again, we would run the app fairly often and pretend to be a user. If anything seemed awkward or strange we would change it

- Testing the layout especially was very repetitive, since the layouts had to be rather precise for the gift images on the main screen. A lot of trial and error was needed to position and scale the images so that they didn't seem out of place.
- The images we used were made specifically for this project- we ended up testing several different image layouts and played around with the color and such quite often. We would run the app and check that the images we used fit well with the lists and buttons, etc that we put on the screen. We ended up having to update the background images often to make room for everything. We also went through several different designs for some of the menu drawer items- especially the gift shop. We had to make sure the icon was sufficiently telling the user what it led to, but also be appealing to look at, as well as fit general vibe of our app.
- For testing the calendar, I (Crystal) used my own Google Account. The Calendar API would only work on my computer since my partner's didn't have the signing key. I also uninstalled the app and reinstalled it to make sure we had the correct permissions, and also tested it without internet.

Usage

- You need a Google account with Google Calendar to sync with the tasks. A UVA gmail account should work fine.
- We pre-loaded the app with almost enough panda points to buy everything, to ease testing. A real version of this app would start at 0.
-

Lessons Learned

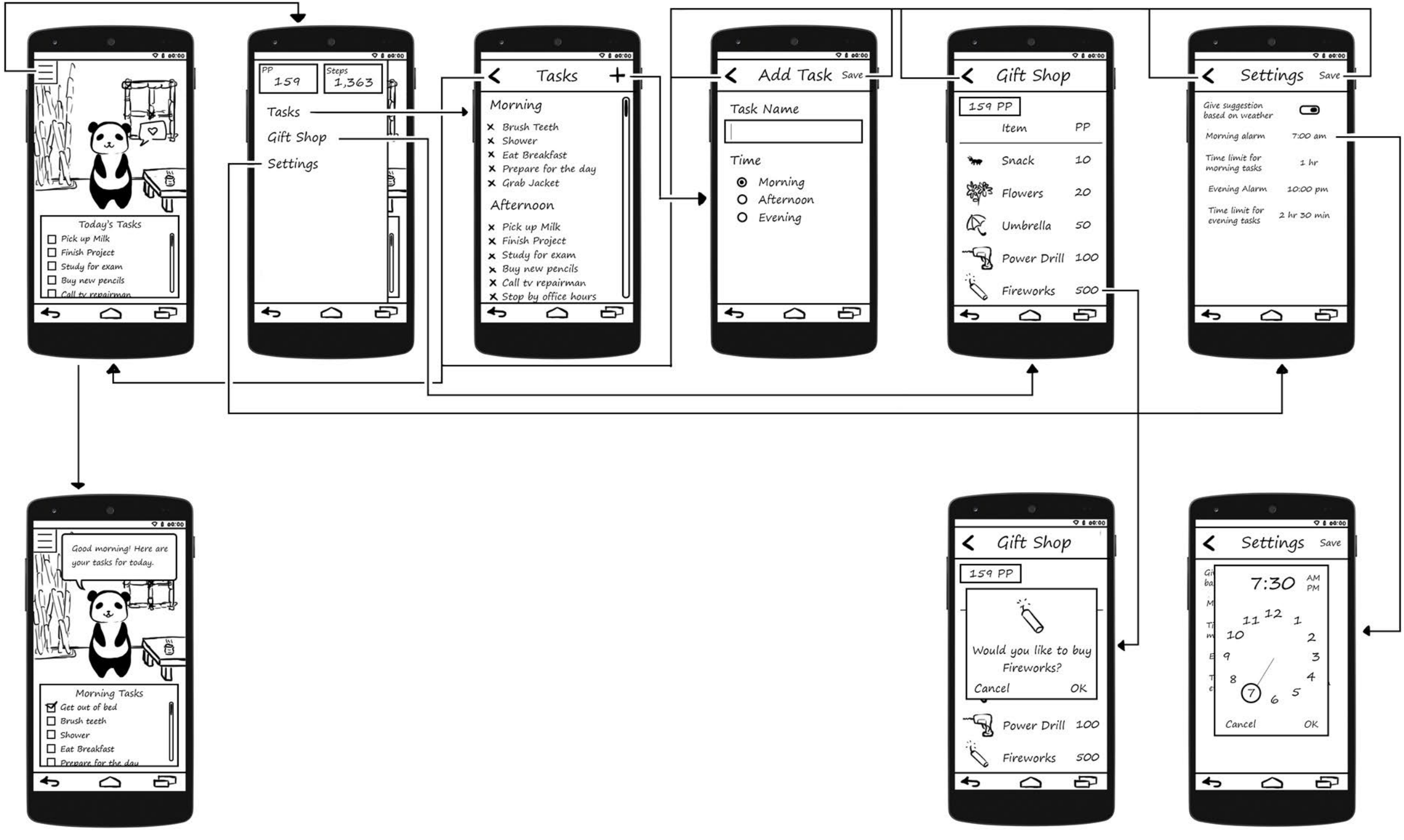
While working on this project, one of the things we've learned was the importance of planning ahead, and how often things don't go according to plan. Even with our wireframe and self made schedules to finish certain screens or functions by a date, we didn't get everything done gradually like we planned. Many outside factors prevented us from properly working on the project until the due date rolled around. Despite this we still got a fair amount done, but if we were perhaps more realistic with ourselves about our motivation levels, we would've set more doable goals.

Another thing that we found interesting was that at first, we were excited about our idea, but when we started building it no longer felt like a fun project. Since in the beginning all we did was build the activities and making sure they didn't crash or do anything strange, it wasn't very exciting. More recently, however, while working on the project we found that it wasn't so bad. THe most rewarding part of building an app is the finished product, and the path that it takes to get there is likely less fun, but something that needs to be pushed through.

In terms of learning how to implement everything we planned, we've learned that the official documentation for an API is often not sufficient to generate working code, since the example code is always basic and lacking in troubleshooting help. Frequently, stackoverflow posts point to important parts of the documentation that would otherwise be difficult to find. Being

conscientious about choosing which stackoverflow posts to implement is also important, because some answers are far more comprehensive than others.

Lastly, we learned how manageable learning how to code on your own really is. It certainly takes a lot of time, moreso than perhaps being walked through a process in an academic setting, but we gained a lot of knowledge and skill just from sitting down, being patient, and getting a lot of feedback from the Android Studio IDE and Google search results. Good documentation is also absolutely essential for anyone to use your code, because otherwise, even if the code is amazing, no one will use it because no one will take the time to try to figure it out.


Today's Tasks
Pick up Milk
Finish Project
Study for exam
Buy new pencils
Call tv repairman
PP
159
Steps
1,363
Tasks
Gift Shop
Settings
Tasks
Morning
Brush Teeth
Shower
Eat Breakfast
Prepare for the day
Grab Jacket
Afternoon
Pick up Milk
Finish Project
Study for exam
Buy new pencils
Call tv repairman
Stop by office hours
Add Task
Save
Task Name
Time
Morning
Afternoon
Evening
Gift Shop
159 PP
Item
PP
Snack
10
Flowers
20
Umbrella
50
Power Drill
100
Fireworks
500
Settings
Save
Give suggestion based on weather
Morning alarm
7:00 am
Time limit for morning tasks
1 hr
Evening Alarm
10:00 pm
Time limit for evening tasks
2 hr 30 min
Good morning! Here are your tasks for today.
Morning Tasks
Get out of bed
Brush teeth
Shower
Eat Breakfast
Prepare for the day
Gift Shop
159 PP
Would you like to buy Fireworks?
Cancel
OK
Power Drill
100
Fireworks
500
Settings
Save
7:30
AM
PM
Cancel
OK