A SLEEP TRACKING APP FOR A BETTER NIGHT'S REST

ABSTRACT:

Health-focused apps have gained increased traction in people's daily lives, from fitness to mood measurement, smoking cessation, diet monitoring or sleep tracking. Scientific studies assess assumptions, scientific grounding, user experiences and ethical issues of such apps, across multiple disciplines. We examine major issues highlighted in studies of sleep tracking apps, to orient designers in this diverse scientific landscape. From measurement accuracy to privacy issues, behavior change models, healthist discourses and user experiences, reviewing the state-of-the-art literature on sleep tracking apps indicates major design choices to inform future solutions.

INTRODUCTION

The vast growth of advancements in Technology has pushed more and more people to incorporate them into their daily life. An industry that has faced a significant disruption due to this is the health and wellness industry. As smartphones are now much more than a phone and wearable technologies reinforce their use, healthcare-related mobile apps are becoming rapidly popular. Healthcare companies and wellness organizations one by one, adapt to this change and try to provide a variety of health-related services to the users of smartphones, tablets, or PCs.

The Client

The National Wellness Institute is an organization founded in 1977. It offers certification and training for wellness professionals, providing them with frameworks and tools that they can use to help clients achieve their wellness goals.

Even though NWI has numerous years of experience in the wellness field, its program has been slow to catch up with technology. They have seen a substantial drop in memberships and want to find a way to add value to their members.

To do this they have decided to focus on two things:

1. Create a set of digital wellness tools for Wellness coaches
2. Update their image — create a new visual system that reflects their innovative and refreshed approach to wellness.

The Challenge

Despite the vast availability of personal metrics and health apps, people continue to struggle to maintain a healthy lifestyle.

We were assigned to conduct user research to understand people's relationship with mental, physical, and emotional well-being in order to develop a tool that will drive them to action.

The National Wellness Institute wanted us to reimagine how people can adopt and maintain a routine that enhances their well-being.

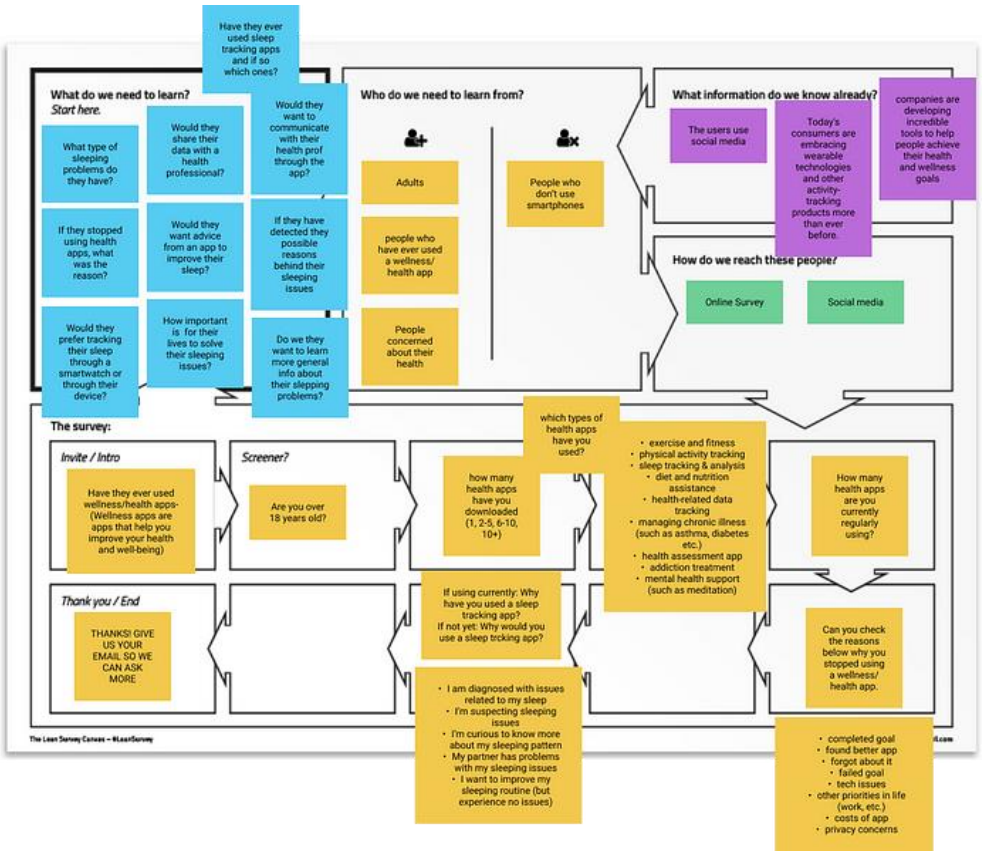
- The tool could be focused in any category that relates to personal wellbeing, such as (but not limited to): exercise and fitness, eating/diet, meditation, time management, etc.
- The only requirement is that it tracks the user's progress and pushes them to commit to a healthier lifestyle.
- The UI should reflect a fresh, updated image.

Scope

- The design should be focused on an MVP, as the app was not supposed to solve all of the wellness issues people have.
- Must-Have:
  Users need to be able to set up their profile to include important information relevant to their goals
  Users need to be able to set goals and track their progress
  Users need to be able to share their stats with their wellness coaches
- Nice to Have:
  Educational component: find a way for users to stay informed throughout the process and understand why they are doing these things, and how it will affect their well-being.
- **Competitive Analysis**
- Initially, we needed to do some market research so that we can identify which would be our direct and indirect competitors. We collected the most used sleep tracking apps like Sleep Cycle, Sleep++, Sleep Score, and Snore Lab, but also apps that track multiple health data like MiFit and Fitbit. After conducting Feature and Brand Analysis, we were ready to do the market positioning of our app. It seems that most apps related to sleep habits offer just one feature and that is mainly tracking sleep.
  Our goal would be to design an app that would provide help to the users so that they can analyze their sleep and improve their sleep habits.

User Research

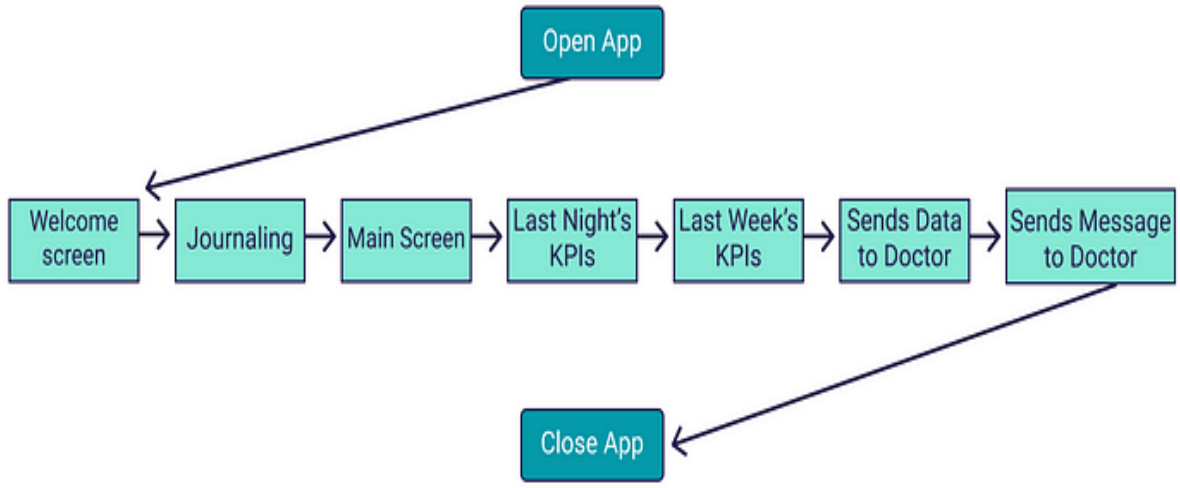
- Our next step was to do user research and we decided to do it with a survey and user interviews. We prepared for the Survey by working on a Lean Survey Canvas which helped us find the right questions for the target audience.



Information Architecture

By testing our Concept Sketches we finalized Jeanine's Happy Flow after using the sleep tracking app for one week.

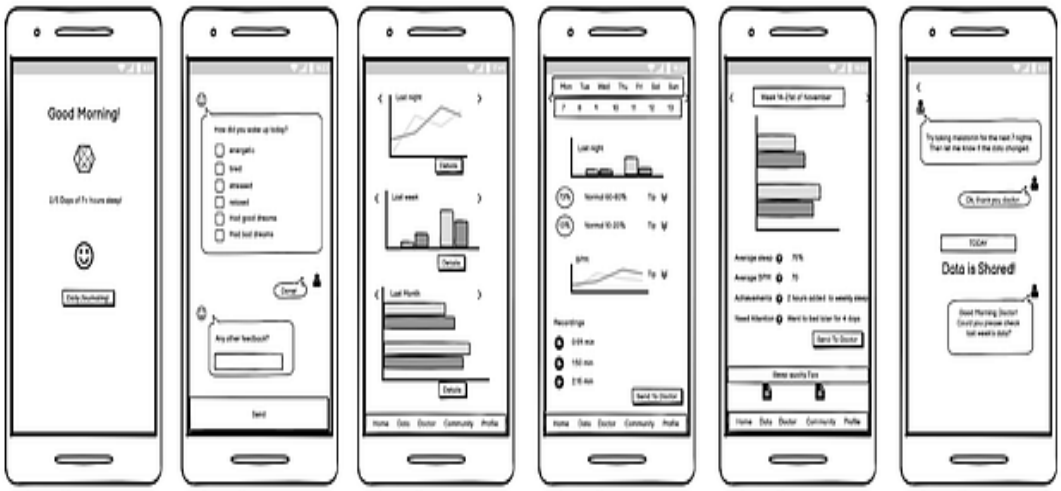
After waking up, Jeanine would open the app (which is also connected to her smartwatch) and she would see a Welcome screen informing her about the progress of her goal and suggesting her to her daily Journaling. Then she would go to the Journaling screen where she could add info about her last night's sleep. Then she would move on to the Main Screen where she would find the summary of her Sleep Data so she will decide to find more details about Last night's sleep and she will navigate to the dedicated screen. To have a more coherent picture of her sleep, she will seek the KPIs of Last Week's sleep and she will share the data report with her doctor through the dedicated chatting screen, from where she will also send them a clarifying question about her sleep cycle.



Jeanine's Happy Path User Flow

Interaction Design

We started giving life to our ideas by incorporating the user flow in a Lo-Fi Prototype for an Android sleep tracking mobile app.

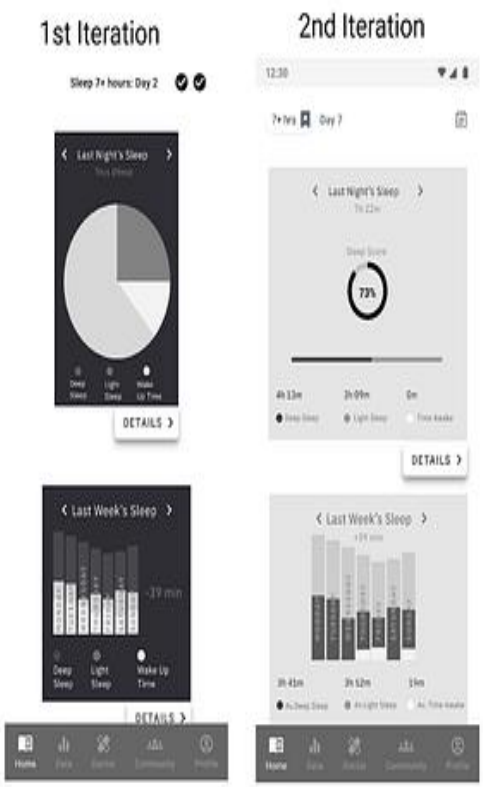


Then we started working on the first iteration of the app's Mid-Fi Prototype which we also tested with 5 Users. The Usability Testing process provided us with very important insights and helped us understand what kind of struggles the users were facing while navigating through the prototype. Based on their feedback we adjusted some features and generated the second iteration of the Mid-Fi prototype.

According to 3 users, the Journaling screen didn't feel intuitive or personal and they would prefer doing their sleep journaling on a screen that resembles a diary more realistically. So we decided to improve the screen by making it more personal and inviting. The chatbot dialog with the checkboxes was replaced by interactive mood buttons that Jeanine would use to document her waking mood, while should could add customized moods and write down her feelings and ideas.

Journaling screen

Although 4 users found the app organized and clean 3 of them, expressed the need to see the sleep KPIs on the main screen in a more structured way. For them, the screen seemed overcrowded and hard to read. That is why in the second iteration we improved the data visualization by giving a better summary of the data of Last Night, Last Week, and Last Month. Jeanine could navigate to the screen quickly inspect her sleep summary(Deep- Light Sleep, Time Awake for each night, or Average Sleep for each week or month) and if needed she could click on the details button and navigate to the page with the complete documentation.



Another screen that we found opportunities to improve was the Weekly Data screen. During the Usability Testing the users shared with us that they would prefer observing all the types of sleep data holistically, and especially on the Weekly Sleep Data it would be more convenient for them to find not only the sleep duration but also the heartbeat rates and the daily mood documentation. We understood that this would help not only the users to have a better picture of their sleep cycles but also their doctor in case they would like to share this data with them. The optimized weekly data screen now included the daily sleep chart with the daily moods that Jeanine had selected along with her daily heartbeat rates. By scrolling down Jeanine would be able to detect her goal achievement and also find tips on how to improve her sleep cycle.

Weekly Data

Usability Testing

5 Users

4 users found the app clean and organised

3 users would like to see the Heart and Sleep data combined

3 users would prefer a diary journaling method instead of a chat

"There is not a lot of info! It's nice and clean."

"I would like to hear if I am talking in my sleep."

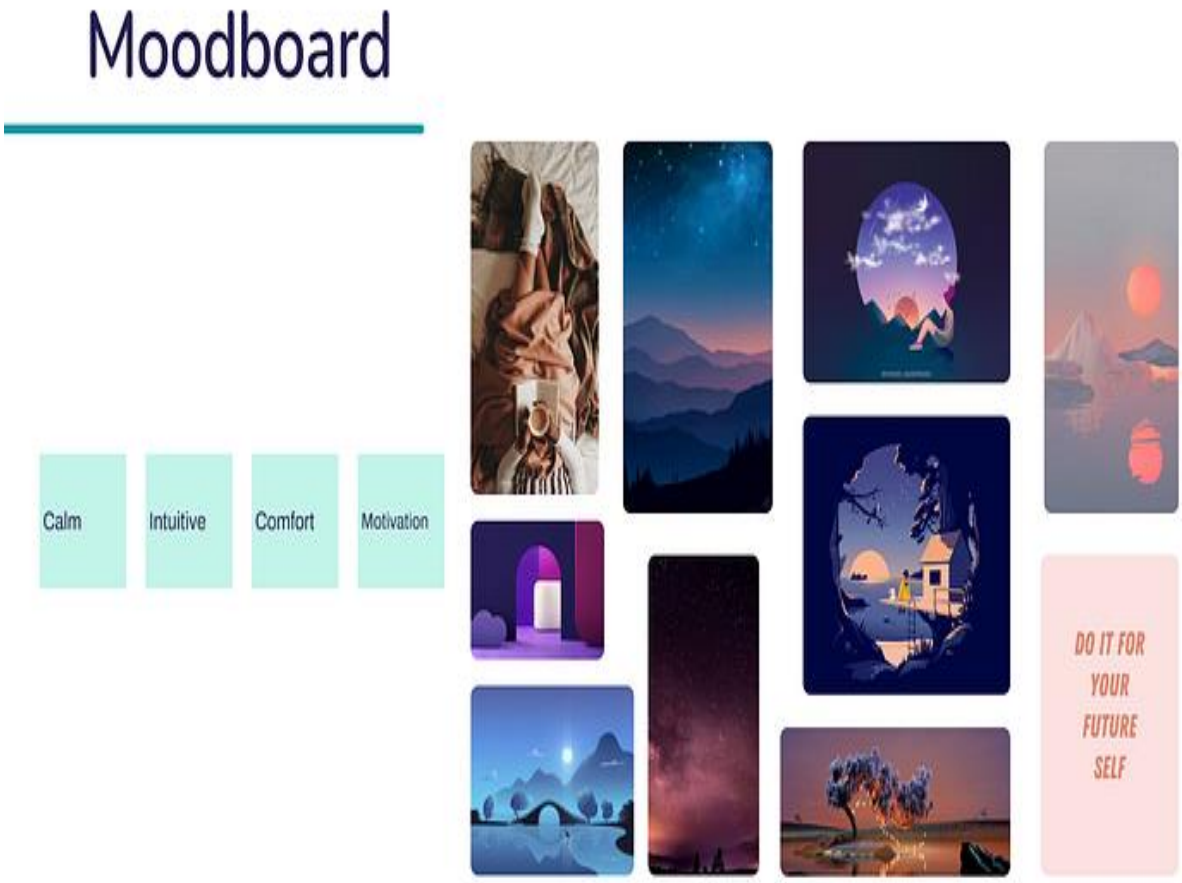
"I would like to see together sleep and heart beat data."

"I would combined the journaling input with sleep data"

"Don't like the chat more like a journal would be better"

Visual Design

Since the interaction design process was completed we moved on to the visual design. The app's Brand Attributes would be Calm, Intuitive, Comfort and Motivation. Based on these attributes we composed a Moodboarad which we also tested with various users, and were happy to see that they would give the same attributes to the board.



We decided that the app should follow the moments of the day while the user would use it and it would have both a Dark and Light mode. However, since we were designing the MVP, we used the dark mode color scheme. We wanted to help Jeanine wake up gently and also easily fall asleep at night, without any intense light from her screen. With the colors we selected we wanted to imitate the light of dawn and sunset and also to respect the eyes of the users, by providing them an accessible interface.

We took into consideration to also include the necessary Multistates as we wanted to provide to Jeanine an app that is intuitive, and that she doesn't need to learn to use it again and again if she has paused its use for some time.

CODE:

Top of Form

```
package com.app.joe.mwsleeptracker;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
/**
 * AppSettingsActivity
 *
 * This activity is displays the currently selected MetaWear board (if one is selected)
 * and allows the user to select a different one.  This activity will call the MWScanActivity
 * to retrieve the MAC of the MW board.
 * The selected board MAC is written as a shared preference.
 */

public class AppSettingsActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_app_settings);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
```

```
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        PrefManager.Init(this);

        Button btnSelectDevice = (Button)findViewById(R.id.btnSelectDevice);
        btnSelectDevice.setOnClickListener(
                new Button.OnClickListener() {
                    public void onClick(View v) {
                        Intent intent = new Intent(AppSettingsActivity.this, MWScanActivity.class);
                        startActivityForResult(intent, 1);
                    }
                }
        );

        TextView textViewSelectedDevice = (TextView)findViewById(R.id.textViewSelectedDevice);
        textViewSelectedDevice.setText("Selected Device: " + PrefManager.readMACAddress());
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        //Send back MAC when Activity is closed.
        if (requestCode == 1) {
            if(resultCode == Activity.RESULT_OK){
                String mac=data.getStringExtra("MAC");

                PrefManager.writeMACAddress(mac);

                TextView textViewSelectedDevice = (TextView)findViewById(R.id.textViewSelectedDevice);
```

```
            textViewSelectedDevice.setText("Selected Device: " +
PrefManager.readMACAddress());
        }
        if (resultCode == Activity.RESULT_CANCELED) {

        }
    }
  }
}
```

Sleep entry:

```
package
com.app.joe.mwsleeptracker;

                                    import java.sql.Date;
                                    import java.sql.Timestamp;
                                    import java.util.Calendar;

                                    /**
                                     * Class SleepEntry
                                     *
                                     * Sleep data is entered into a instance of this class and
                                    is sent to the
                                     * dbhandler to be inserted into the database
                                     */
                                    public class SleepEntry {
                                        private long id;
                                        private long logdatetime;
                                        private int sleepState;

                                        public long getId() {
                                            return id;
                                        }
```

```
    public void setId(long id){
        this.id = id;
    }

    public long getLogDateTime() {
        return logdatetime;
    }

    public void setLogDateTime(long logdatetime){
        this.logdatetime = logdatetime;
    }

    public int getSleepState() {
        return sleepState;
    }

    public void setSleepState(int sleepState) {
        this.sleepState = sleepState;
    }

}
```

Sleep Log:

```
package
com.app.joe.mwsl
eeptracker;

import android.graphics.Color;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
```

```
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.TextView;
import org.achartengine.ChartFactory;
import org.achartengine.GraphicalView;
import org.achartengine.model.TimeSeries;
import org.achartengine.model.XYMultipleSeriesDataset;
import org.achartengine.renderer.XYMultipleSeriesRenderer;
import org.achartengine.renderer.XYSeriesRenderer;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

/**
 * SleepLogActivity
 *
 * This activity is used to view the sleep log.
 * Selecting a date from the calendar picker will execute a SQL query that returns
 * the sleep information for the date selected.  The sleep state transitions
 * are shown on a graph at the bottom of the page.  The activity is started with
 * the current date as a default value.
 */
public class SleepLogActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sleep_log);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        //Set the current date as a default value
        TextView tvDate = (TextView) findViewById(R.id.tvDate);
        Date date = new Date();
        SimpleDateFormat df = new SimpleDateFormat("MM-dd-
yyyy");
        String formattedDate = df.format(date);
        tvDate.setText(formattedDate);

        //Create a listener that will detect when the date value
        //has been changed.
        tvDate.addTextChangedListener(new TextWatcher() {
           public void afterTextChanged(Editable s) {
              //Query the database and genereate the graph
              generateGraph();
           }

           public void beforeTextChanged(CharSequence s, int start,
                                 int count, int after) {
           }

           public void onTextChanged(CharSequence s, int start,
                              int before, int count) {
           }
        });

        //Query the selected date and display the graph
```

```
        generateGraph();
    }

    private void generateGraph(){
        //Generate the graph of the sleep data for the selected date
        Calendar calendarStart = null;
        Calendar calendarEnd = null;

        //Get date from the text view
        SimpleDateFormat df = new SimpleDateFormat("MM-dd-
yyyy HH:mm");

        TextView tvDate = (TextView)findViewById(R.id.tvDate);
        String searchDateString = tvDate.getText().toString()+ "
23:59";

        try{
            //Create the start and end dates for the search
            Date searchDate = (Date)df.parse(searchDateString);

            //Starting with the 23:59 on the selected date
            //Subtract 12 hours to create a start date/time
            //Add 12 hours to create a end date/time
            calendarStart = Calendar.getInstance();
            calendarStart.setTime(searchDate);
            calendarStart.add(Calendar.HOUR_OF_DAY, -12);

            calendarEnd = Calendar.getInstance();
            calendarEnd.setTime(searchDate);
            calendarEnd.add(Calendar.HOUR_OF_DAY, 12);
        }
        catch (ParseException e){
```

```
        }

        //If the dates have been successfully create then query the
database
        if (calendarStart != null && calendarEnd != null){
            int[] sleepStatus = null;
            Date[] sleepDate = null;

            DBHandler dbHandler = new
DBHandler(SleepLogActivity.this);
            SleepLog sleepLog =
dbHandler.getSleepLog(calendarStart.getTimeInMillis(),
                    calendarEnd.getTimeInMillis());

            sleepStatus = sleepLog.getSleepStatus();
            sleepDate = sleepLog.getSleepDate();

            //If there is no sleep data logged for the given date, display
a message and
            //hide the textviews and graph
            if (sleepStatus == null || sleepStatus.length < 1){
                TextView tvTotalAsleep =
(TextView)findViewById(R.id.tvTotalSleep);
                tvTotalAsleep.setText("There is no log data for the date
selected");

                TextView tvTotalAwake =
(TextView)findViewById(R.id.tvTotalAwake);
                tvTotalAwake.setVisibility(View.GONE);

                TextView tvTotalRestless =
(TextView)findViewById(R.id.tvTotalRestless);
                tvTotalRestless.setVisibility(View.GONE);
```

```
            LinearLayout layout = (LinearLayout)
findViewById(R.id.chart);
            layout.setVisibility(View.GONE);
        }
        else
        {
            //If there is data
            int[] margins = {100, 150, 100, 25};

            //Create the chart
            GraphicalView mChart;
            TimeSeries tsSleepQual = new TimeSeries("Sleep
Quality");

            for(int i=0; i<sleepDate.length;i++){
               tsSleepQual.add(sleepDate[i], sleepStatus[i]);
            }

            XYMultipleSeriesDataset mDataset = new
XYMultipleSeriesDataset();
            mDataset.addSeries(tsSleepQual);

            XYSeriesRenderer mSleepRenderer = new
XYSeriesRenderer();
            mSleepRenderer.setColor(Color.GREEN);

            XYSeriesRenderer.FillOutsideLine fill = new
XYSeriesRenderer.FillOutsideLine(XYSeriesRenderer.FillOutsi
deLine.Type.BOUNDS_ABOVE);
            fill.setColor(Color.GREEN);
            mSleepRenderer.addFillOutsideLine(fill);
```

```
        XYMultipleSeriesRenderer mRenderer = new
XYMultipleSeriesRenderer();
        mRenderer.addSeriesRenderer(mSleepRenderer);
        mRenderer.setLegendTextSize(48f);
        mRenderer.setFitLegend(true);

        mRenderer.setYLabels(3);
        mRenderer.setXLabels(4);
        mRenderer.setYAxisMin(0);
        mRenderer.setYAxisMax(10);

        mRenderer.setMargins(margins);
        mRenderer.setYLabelsPadding(80f);
        mRenderer.setLabelsTextSize(48f);

        mRenderer.addYTextLabel(0, "AWK");
        mRenderer.addYTextLabel(5, "RST");
        mRenderer.addYTextLabel(10, "ASLP");

        mRenderer.setPanEnabled(true,false);
        mRenderer.setZoomEnabled(false, false);
        mRenderer.setChartTitle("");
        mRenderer.setYTitle("");
        mRenderer.setXTitle("");
        mRenderer.setGridColor(Color.WHITE);
        mRenderer.setBackgroundColor(Color.BLACK);
        mRenderer.setApplyBackgroundColor(true);
        mRenderer.setShowGrid(true);

        mChart = ChartFactory.getTimeChartView(this,
mDataset, mRenderer, "MM-dd HH:mm");

        //Add the chart to the layout
```

```
            LinearLayout layout = (LinearLayout)
findViewById(R.id.chart);
            layout.addView(mChart);
            layout.setVisibility(View.VISIBLE);

            //Update textviews
            TextView tvTotalAwake =
(TextView)findViewById(R.id.tvTotalAwake);
            tvTotalAwake.setText("Time Awake (mins): " +
String.format("%.0f",
(double)((double)sleepLog.getTotalAwake() / 1000) / 60));
            tvTotalAwake.setVisibility(View.VISIBLE);

            TextView tvTotalSleep =
(TextView)findViewById(R.id.tvTotalSleep);
            tvTotalSleep.setText("Time Asleep (mins): " +
String.format("%.0f", ((double)sleepLog.getTotalAsleep() /
1000) / 60));
            tvTotalSleep.setVisibility(View.VISIBLE);

            TextView tvTotalRestless =
(TextView)findViewById(R.id.tvTotalRestless);
            tvTotalRestless.setText("Time Restless (mins): " +
String.format("%.0f", ((double)sleepLog.getTotalRestless() /
1000) / 60));
            tvTotalRestless.setVisibility(View.VISIBLE);
        }
    }
  }
}
```

OUTPUT SCREEN

Loading- Progress Ring

Blank-First View

Alert/Notification

As soon as the Hi-Fi Prototype was ready, we conducted a Desirability testing so to understand how the users perceive the app and what feelings it causes them.

We tested the prototype with 5 users and we were glad to see that all of them were feeling that the app is clean and comforting. For most of the users, the app felt personal and made them feel happy. Everyone expressed the desire to use it and that meant that our goal was achieved.

Conclusions

It was very interesting to discover that when users are using a health app they want to feel that it is more personal and not like they are using a cold automated device. For them, it was crucial to feel that their sleep is automatically tracked but that they also have the opportunity to provide their own input. Only then they would feel that their data has value for their health.

As designers, it was a great reminder to discover that users with sleep issues need clean and organized info as this would cause them less stress. The fast pace routines usually affect them not only during the day but mainly right before their night sleep so it was crucial for them to have an app that would ease this disturbance.