Functional and Technical Specifications Document

WeathoMeter

Project Name: WeathoMeter

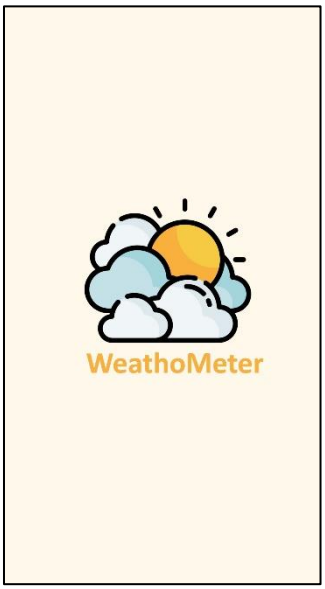
Group Members:

1. **Krishna Parmar (200504981)**
2. **Sajan Tamang (200507132)**

Functional Requirements

WeathoMeter is a Weather Application. The purpose of the project is to develop an application in which users can view current temperature and other weather-related details of a particular location. Along with this, user can save the location for future use. User can get notified to check weather on selected time for selected location.

1. **Splash Screen:**
   Initially, splash screen will be loaded which will include the logo of the application.

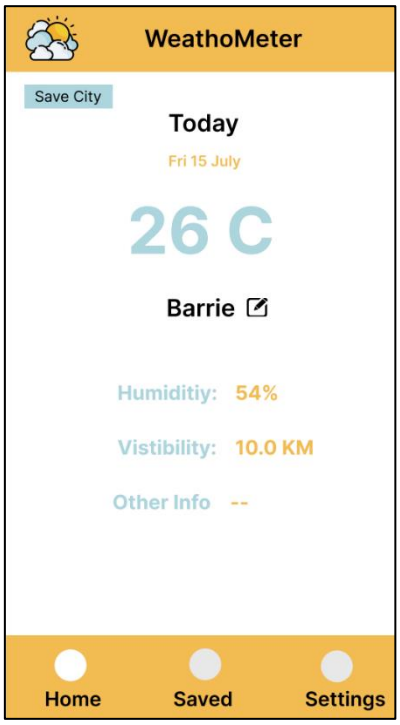


2. **Home Screen:**

   In this screen, User will be able to see the weather Information of Of a particular Location.

   Using Pencil button, user can change the city.

   If User wants to save the city for later use then it would be able to saved using SAVE button.



3. **Saved City:**
   In this screen, User will be able to see the saved city and it's weather details.

   Just in case, If user wants to delete the city then they can do that using Delete Button.

4. **Settings Screen:**

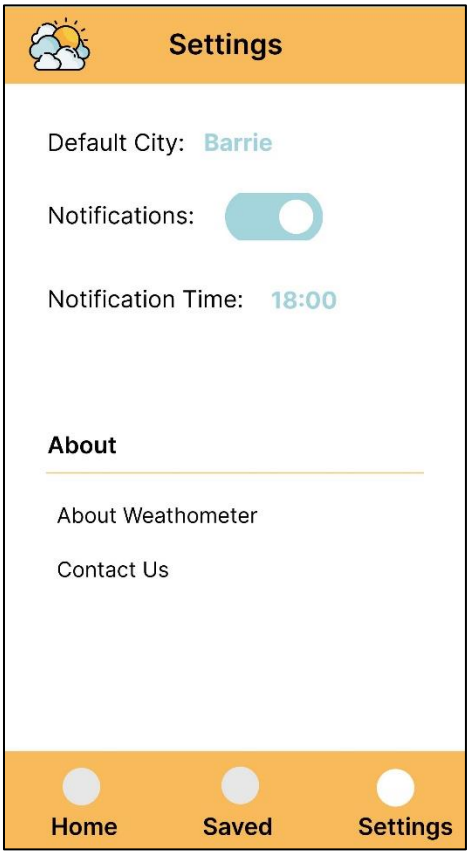
   In bottom navigation, when user clicks on Settings menu,

   User will be redirect to 'Setting Screen'.

   This screen allows users to enable or disable notification alert And allow to change the time of notification.

   User can save the default city as well in this Screen

   About section contains the information about Weathometer app and contact information.



Technical Specifications

1. **Home Screen:**
   Model class: City
   Variables for City class:
   - CityId which is an integer
   - CityName which is a string

   Functions:
   fetchWeatherInfo() -> this function queries and returns all the weather Information from https://openweathermap.org/ API.

   SaveCity() -> this function will be used to save the City name in SQLite

2. **Saved City Screen:**
   Model class: City
   Variables: CityName is a string
   CityId is an integer
   Functions:
   fetchData() -> To Get the Weather Info of the City from the API.
   deleteCity() -> to delete the City from the list as well as local database
   The data are fetched from the local database and contains functions to fire the queries for delete the City.

3. **Settings screen:** It has option to send daily notification in which user can turn it off/on and set the time to send notification. In this Screen User can set the Default city which will be useful to see the temperature in HomeScreen. If it is not changed by the user then by default the city will be Barrie.
   Functions: setTimer() -> to set timing of notification
   SetDefaultCity() -> to Set the city in the local database

Database: Sqlite

Table Name: City
CityId (Integer) (Primary Key) (Auto_Increment)
CityName (Text)

Table Name: Notification
isNotification (Boolean)
notificationTime (Text)

City table will be use to store details of City.
Notification table will be used to store the details of whether notification is enabled/disabled as well as the time of notification to be send.