DISTANCE CALCULATOR

Complete Design and Tech Documentation

Table of contents

Application Summary

The Distance Calculator application allows users to calculate distances between two locations.

The distance is calculated for air line and for road also, including car, bike, and walk distances – if available. Road distance is also calculated for travel time.

The Location Selection can be used via selecting from bigger cities from the world.

The Selection can also be made by applying current user location.

There is also an opportunity to open a Map, and mark selections.

The results are shown in a page with nice and fancy animations.

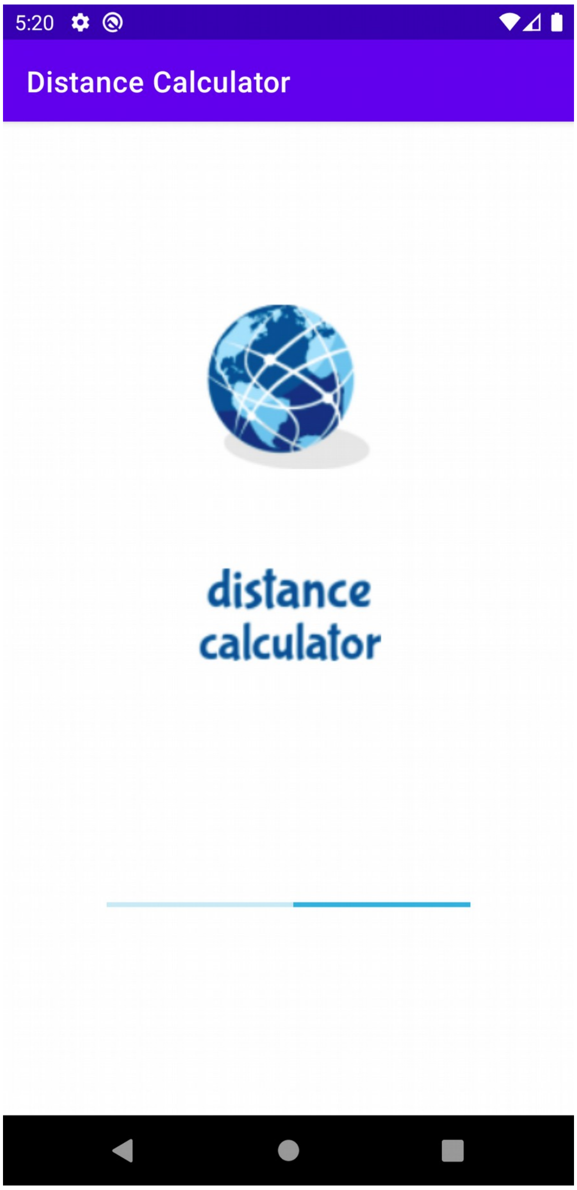
Users are allowed to customize their app by changing Settings.

The language of the application, the render type of the Map, and the units used for calculations are all able to be personalized.

The application supports extended user accessibility by supporting multiple languages (English, Hebrew, Arabic, Hindi).

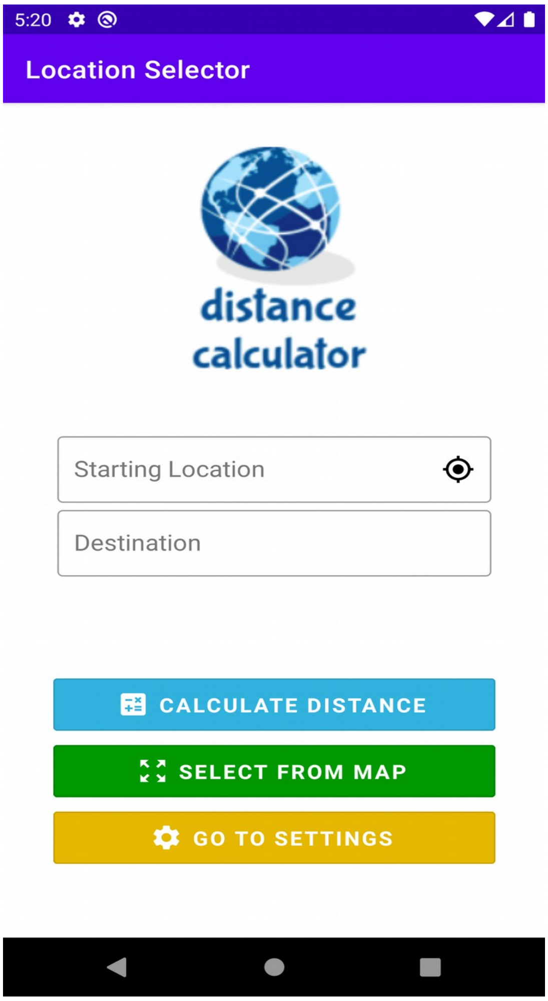
The application can be used without network connection, although some more complex calculations need internet access.

Distance Calculator – Loading Screen



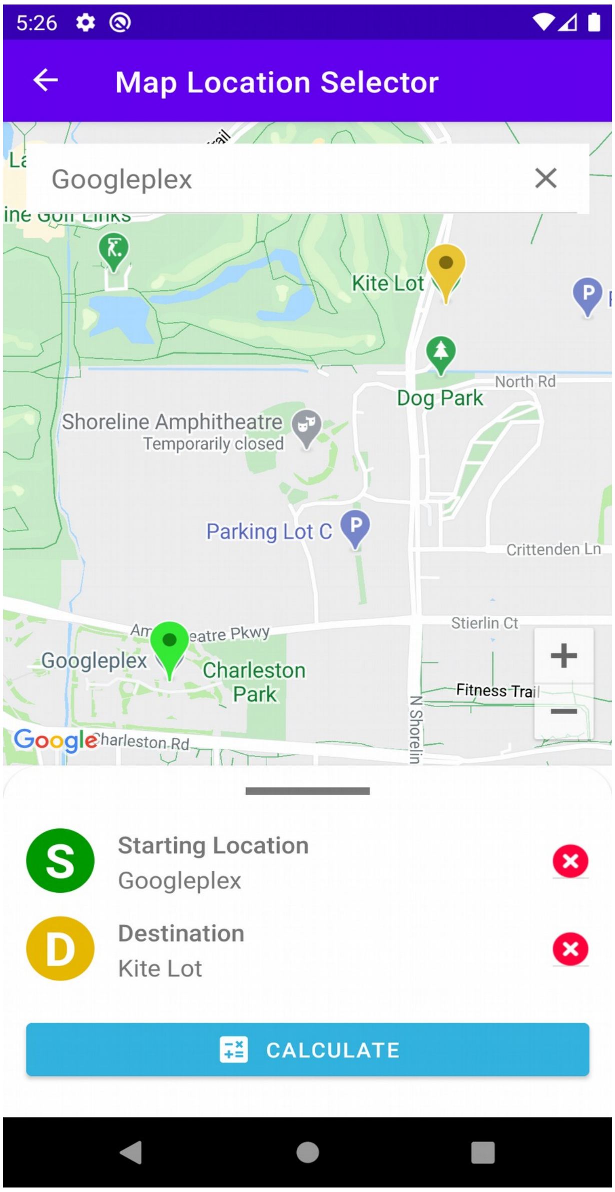
- Initial loading screen before landing to main page
- Contains application logo
- Linear progress bar is shown with ‘Fake Loading’ to indicate loading progress and increase User Experience

Distance Calculator – Selector Screen



- Selector Screen to select destinations
- Alternate app logo in the top
- Two auto-complete Material Text Views for setting start location and destination for calculation
- Auto-complete data is loaded from cache, which is fetched form remote data source
- Target Icon next to starting location selector text view to allow user to select current location when pressing on it
- Current Location feature is only enabled when location is turned on
- If Location is turned off, Target Icon disappears
- Calculate button at the bottom to start calculation
- Button with Custom Icon to indicate Location Selection via Map
- Settings menu option, which will navigate to Setting Screen
- Location Permission is asked when entering the Screen

Distance Calculator – Map Selector Screen



Map Location Selector

Googleplex ✕

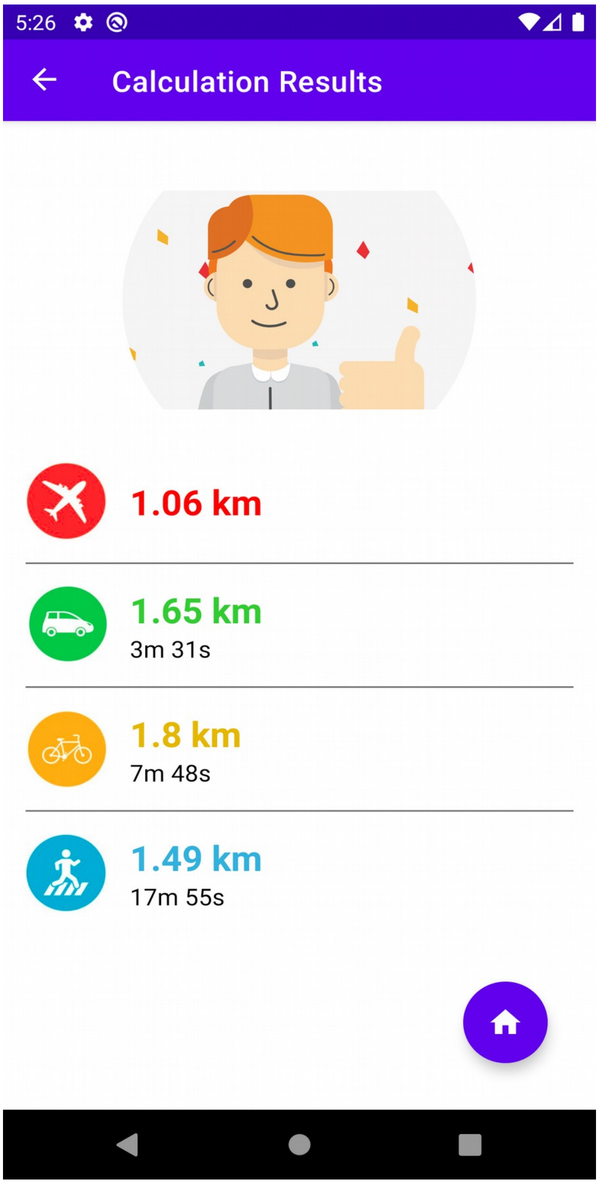
Starting Location
Googleplex

Destination
Kite Lot

CALCULATE

- Map Selector Screen to select destinations from Map
- If Location is enabled, Map is zoomed by start at Current Position
- Can be used without Location enabled, Google is default Starting Point
- User can select Start and Destination Location by Creating Markers on Points of Interests, which are shown in a Bottom Sheet
- Selections can be removed by clicking the Cancel Icon next to them
- Bottom Sheet can be slid by user, and it also automatically expands when a Selection is made.
- Bottom Sheet is automatically collapsed, when there is no Selection
- Search View on the top, to search for Points of Interest
- Calculate Button for triggering calculation and navigating to Results Screen

Distance Calculator – Results Screen



5:26

Calculation Results

1.06 km

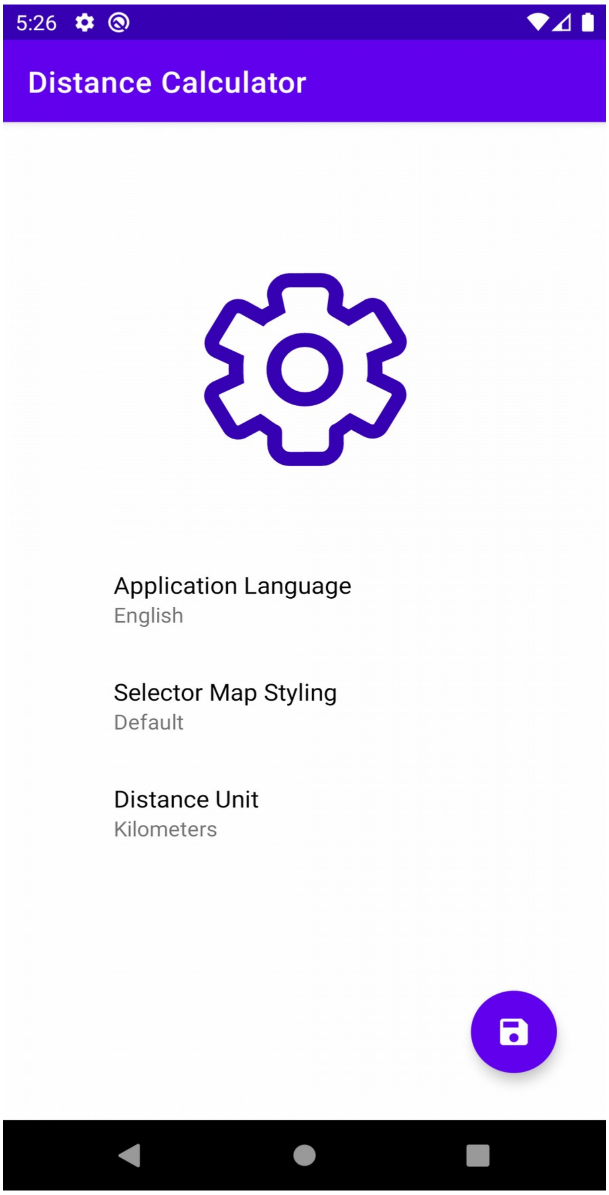
1.65 km
3m 31s

1.8 km
7m 48s

1.49 km
17m 55s

- Results Screen to show calculation results
- Fancy animation for high quality User Experience
- Animation shows Loading Screen until results are ready
- Results contain two lines (the distance, and the time it takes)
- Upper line shows the distance by air line, and is always calculated
- Road distances (Car, Bike, Walk) are only calculated when there is network connection
- When a bad request is sent (like a distance which cannot be calculated for roads), Bad Request screen is shown, but Airline distance is calculated
- When there is no Network Connection, Disconnected Screen shows
- Floating Action Button in the bottom to navigate back to Selector Screen

Distance Calculator – Settings Screen



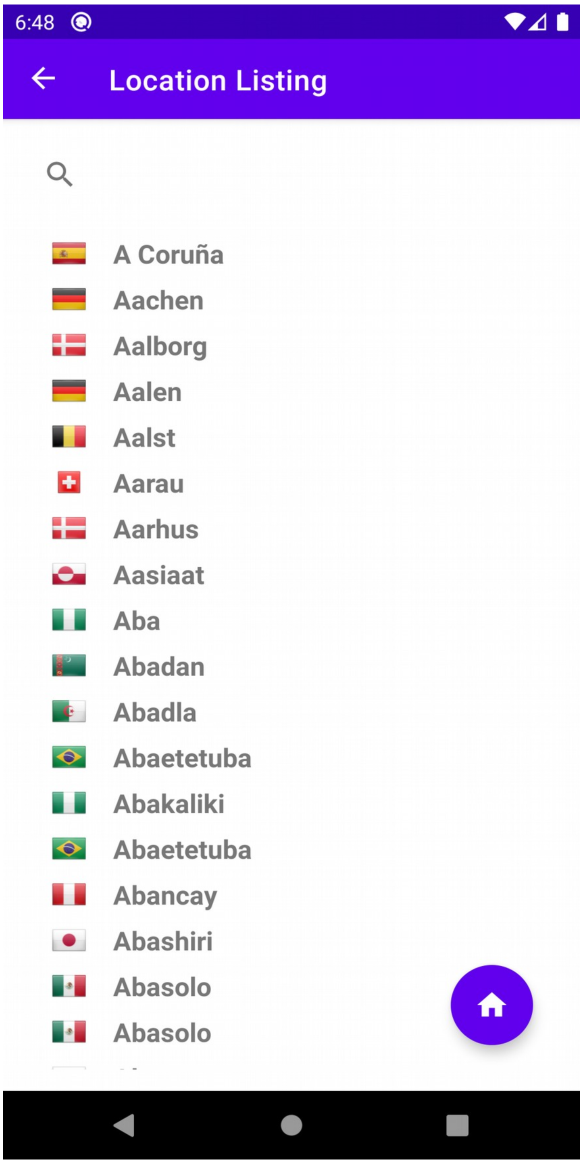
Application Language
English

Selector Map Styling
Default

Distance Unit
Kilometers

- Settings Screen to personalize application for Users
- Multiple Language Support (English, Hebrew, Arabic, Hindi)
- Distance unit selection (km or mile)
- Map Type selection (Default, Hybrid, Satellite, Terrain)
- Save Button in the bottom to Save Selected Settings, and navigate back to Main Screen
- Language change automatically applies to the whole application when saved
- In some cases the Map Selector’s Point of Interests will only be translated to Current Language, when the Application is restarted

Distance Calculator – Location List Screen



A Coruña
Aachen
Aalborg
Aalen
Aalst
Aarau
Aarhus
Aasiaat
Aba
Abadan
Abadla
Abaetetuba
Abakaliki
Abaetetuba
Abancay
Abashiri
Abasolo
Abasolo

- Location Listing Screen to list available locations
- Cities are listed with the respective flag of their Country
- More than 26k items
- Search View to filter results by text the Location Name contains
- When Search View cleared, the list resets itself to show all values
- Home Button in the bottom to navigate back to Main Screen

Technological Stack

Model – View – View Model

Clean and modern application architecture

Kotlin Co-routines

Easily managing asynchronous calls and callbacks

Data Binding

Connecting Views with data needs to be showed

Material Design Components

Best practices for modern User Experience and Design

Navigation Component with Safe Arguments

Advanced navigation with type safe arguments between Views

Object Box (with Live Data)

Super fast local data caching with Live Data extensions

Retrofit (with Live Data)

Network connecting and fetching tool with Live Data extensions

Google Play Services

Map functionality for extended User Experience

Glide Image Loading Library

Loading images easily and effectively to Views

Dagger – Hilt

Dependency Injection at its highest level

Three Ten Back Port

Date and Time management for all API levels

Meeting criteria

Android User Interface/User Experience

Build a Navigable Interface

- ✔ Application includes 5 screens and using both Navigation Component and Intents
- ✔ Navigation Component is used for Fragment based Navigation
- ✔ Bundle is used to store and send data passed between Fragments

Construct interfaces that adhere to Android standards

- ✔ Application uses Constrain Layouts in almost every screen
- ✔ Data collection (list of cities) are shown in Auto Complete Text Views
- ✔ Resource values are stored in their respective XML files
- ✔ Elements inside Constraint Layouts has ID, and constrained both vertically and horizontally
- ✔ List of Locations are loaded into the Recycle View using View Holder Pattern

Animate User Interface components

- ✔ Launch Screen and Settings Screen each contain Motion Layouts with animations
- ✔ Motion Layouts use Motion Scenes
- ✔ Constraints are defined within the scenes

Local and Network Data

Connect to and consume data from remote data source

- ✔ The Application connects to Firebase remotely to fetch list of cities, and it also connects to Open Route Services for road distance calculations. Retrofit library is used for that.
- ✔ Local Model Objects (like City Model) hold fetched data
- ✔ Data fetching is implemented in a background thread using co-routines

Load resources dynamically and on-demand

- ✔ Glide is used to load small country flags next to Locations
- ✔ Placeholder and error images are used for appropriate states
- ✔ Data request is made in background thread

Store data locally on device

- ✔ Shared Preferences are used for saving and loading User Settings
- ✔ Object Box is used to cache Location Data
- ✔ Data stored is accessible through user sessions
- ✔ Data storage operations are performed on background thread
- ✔ Appropriate data types and scopes are used

Android System and Hardware Integration

Using Model–View–View Model architecture

- ✔ Clean architecture is implemented, with Live Data components

Implement logic to handle hardware and system events

- ✔ Orientation changes to be handled via View Models
- ✔ Life cycle events are also handled by View Models
- ✔ Intents are used for handling interaction to and from the application
- ✔ Location Permission is handled and managed by the Application

Utilize system hardware

- ✔ Application utilizes the Location hardware component
- ✔ Run time permission is requested at the time of use
- ✔ Location based functionalities to be available only for granted permissions

Bonus

- ✔ Animations and custom styles are used for a more positive user experience
- ✔ Application utilize locale data storing with local database and Shared Preferences
- ✔ Personal needs to be satisfied with user settings enabled
- ✔ Application has multi language support (also right-to-left language support)
- ✔ Application provides value over a website, since distances can be calculated from the scene
- ✔ Application intuitively guides the user’s behavior

Project Starting Date: 2021. 01. 29.
Project Deadline Date: 2021. 02. 22.

Time spent per day: 2-3 hours

The Design Document is sent with the project on 2021. 02. 19.

Application Developer is:
Károly Ujszászi