# Project Description:

CalcBox is a multi-functional mobile application designed to consolidate a wide range of calculators and converters into a single, easy-to-use platform. It includes number base converters, scientific calculators, unit converters, interest calculators, and more organized in a customizable tile grid interface. Users can personalize the app experience by selecting visible tools, adjusting tile sizes, and enabling themes like dark or light mode. The goal of the project is to improve productivity by minimizing the need to switch between multiple websites and calculator apps, thereby enhancing user convenience and accessibility.

# Requirements Summary:

|  |  |  |
| --- | --- | --- |
| **MINIMUM REQUIREMENTS** | Processor Cores | Dual Core |
| OS | Android 7.0 (Nougat) |
| RAM | 1 GB |
| Storage | 50-100~ MB free space |
| **RECOMMENDED REQUIREMENTS** | Processor Cores | Quad Core |
| OS | Android 8.0 (Oreo) |
| RAM | 4 GB |
| Storage | 200~ MB free space |

Table 1. System Requirements

**Note:** iOS support is not yet available for this application as development and testing were exclusively conducted using Android Studio, which is tailored for Android devices. Compatibility with iOS would require additional development and testing using Xcode on macOS.

# Prototype Description:

The prototype was created using Figma, a widely used interactive prototyping tool that allowed the group to design and simulate the look and flow of the application. Figma was chosen because it enables quick adjustments and allows testers to interact with the UI through shareable links, making distribution and feedback collection easier and more efficient.

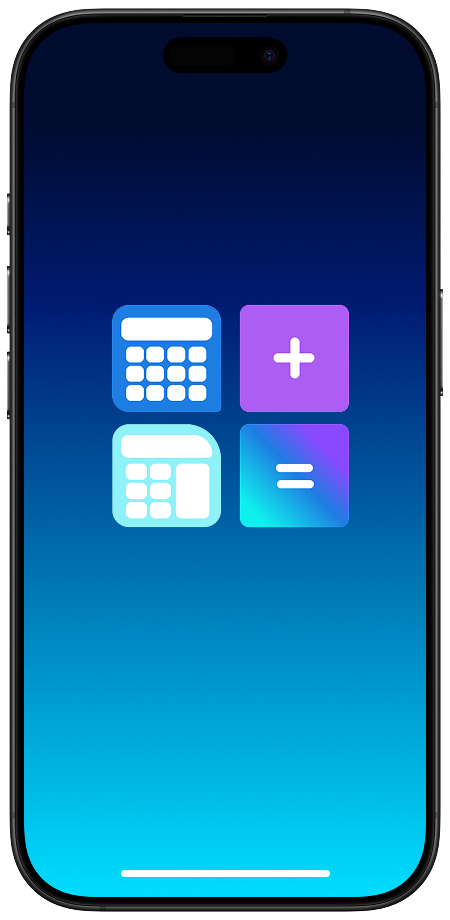
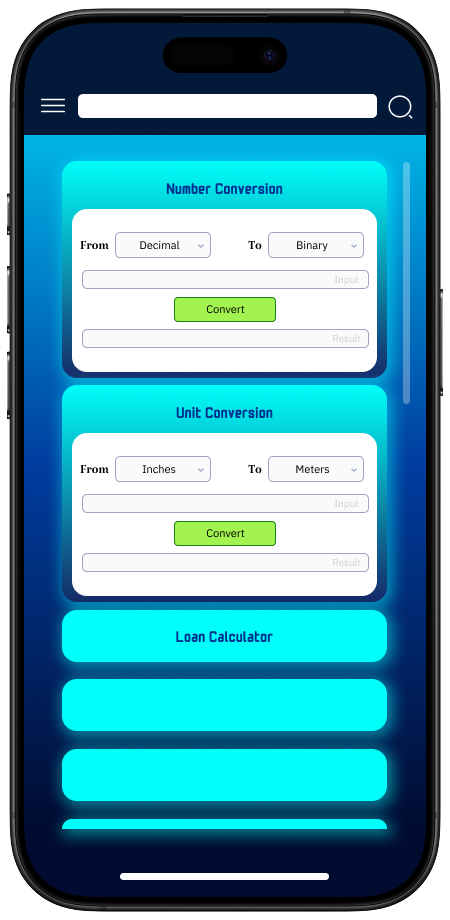
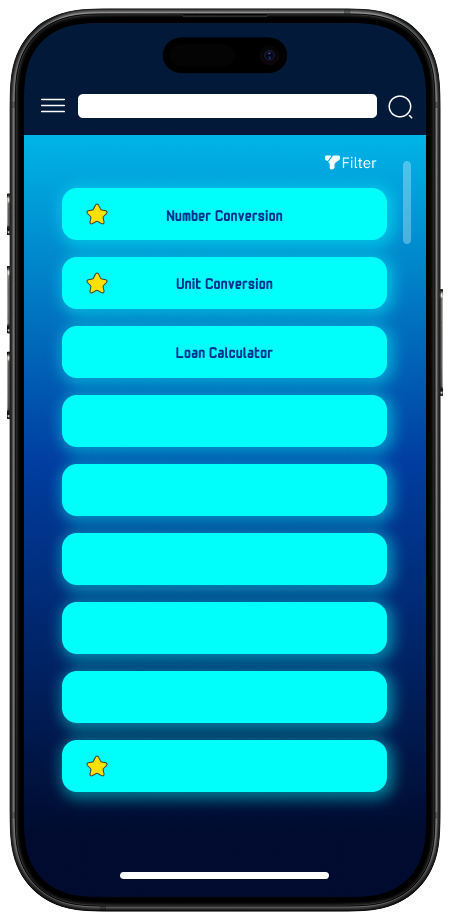
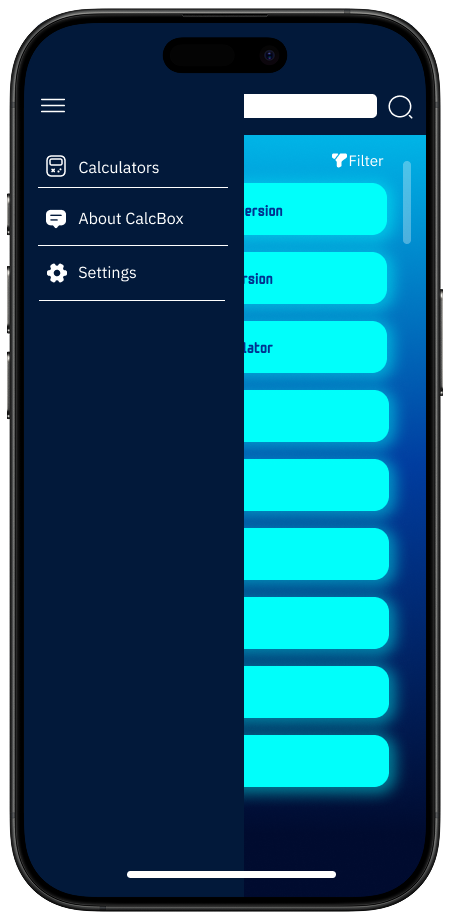
In addition to the Figma prototype, the group also used the actual unfinished version of the working app for testing purposes. This version, although missing the Settings page and some calculator functions, already allowed users to interact with the main features such as selecting and using individual calculators. The working prototype was installed on the testers’ mobile devices directly by the developers to gather more realistic, hands-on feedback. This combination of visual simulation through Figma and real device interaction helped provide a broader perspective on both the design and functionality during the evaluation phase.

# User Scenario:

Mike was working on a home improvement project when he ran into a problem; his countertop measurements were in inches, but all the materials he looked at online were listed in centimeters. He needed to convert 72 inches to centimeters, but he couldn’t remember the exact formula. To make matters worse, his internet connection was slow, making it impossible to quickly search for the conversion.

A while back, Mike’s friend recommended an app called CalcBox, but he hadn’t given it much thought until now. Remembering the app, he quickly opened CalcBox and found the unit converter feature. Without needing the internet, Mike easily converted the measurements and was able to canvas how much wood he needed before he went out to buy the supplies.

# CalcBox Mock-up/Prototype:

The Side Bar is where the user will navigate to the Settings and other features to be added in the future.

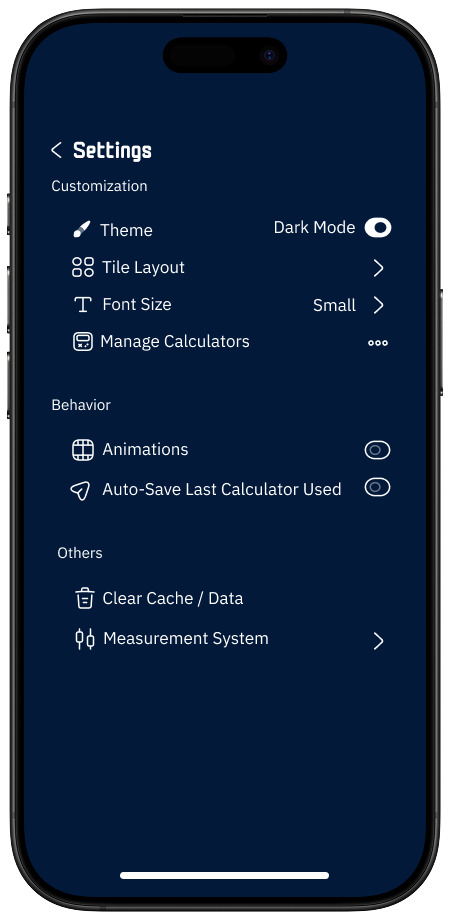
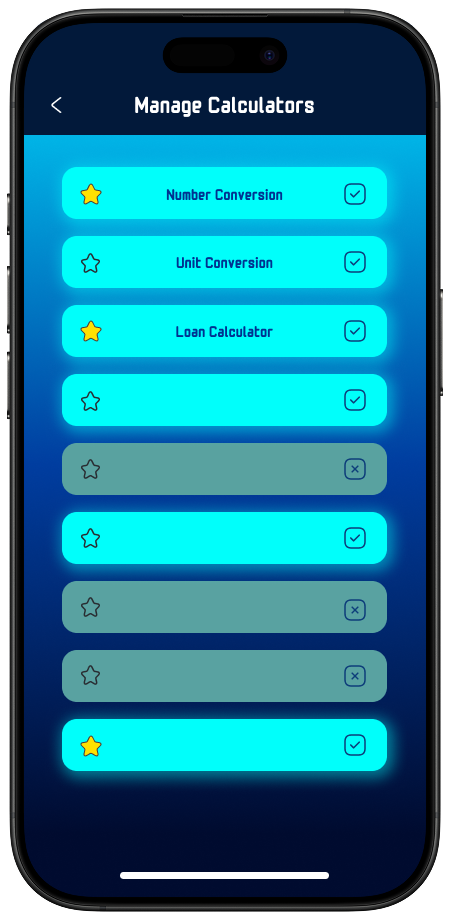
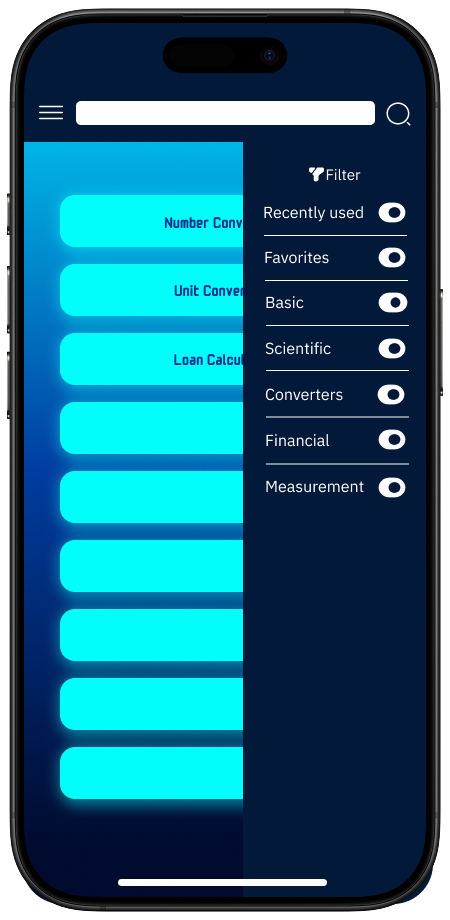
**Splash Screen**

The Main Screen is where the user will be able to navigate to find a calculator they need. Each tile presents a calculator and is opened through a dropdown.

The Splash Screen contains the app’s logo and will be displayed for a few seconds on startup.

**Main Screen**

**Side Bar**

**Settings Screen**

The Manage Calculators screen from the Settings is where the user will be able to select calculators to be shown in the main screen.

The Settings screen is where the user will be able to customize the appearance and behavior of the app by their preferences and manage other aspects.

**Manage Calculators**

**Filter Bar**

The Filter Bar is where the user will be able to filter out calculators to easily find out what they need.

# Prototype Flow:

**Main Screen:**



Figure 2. Entering Prototype



Figure 3. Interacting with Calculator Tiles



Figure 4. Going to Settings

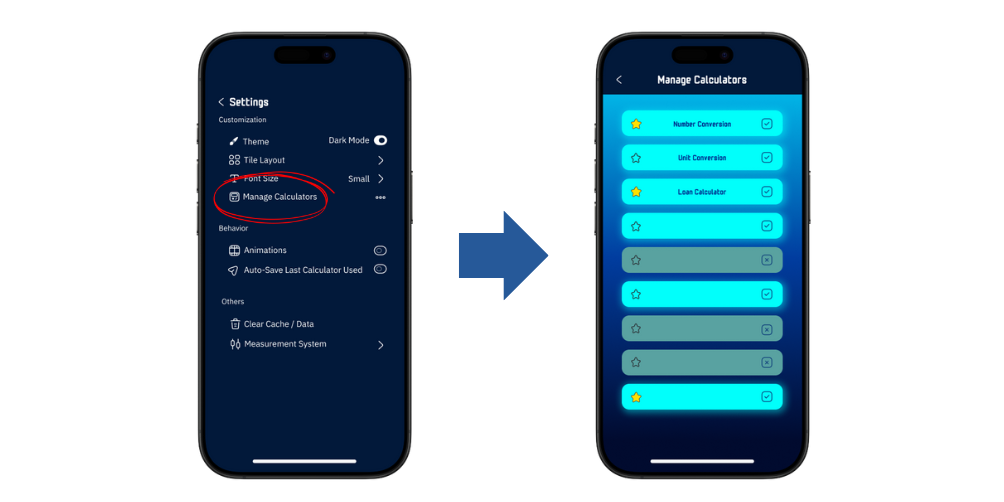


Figure 5. Managing and Selecting Calculators



Figure 6. Filtering Calculators

# Rationale:

The team chose a mobile app prototype using Capacitor and web technologies to maximize compatibility, rapid development, and ease of testing across a wide range of Android devices, as outlined in our requirements. This approach allows us to dynamically render calculator tiles, provide customizable settings (like dark mode, disabling animations, and notifications), and efficiently manage features such as enabling/disabling calculators all within a single-page application for fast, responsive user experience. This design enables thorough evaluation through emulator and real-device testing, user observation, and feedback collection, supporting our usability goals of effectiveness, efficiency, and learnability. While hybrid apps may not match native performance on low-end devices, our modular and configurable UI, along with performance options, ensures the prototype remains accessible and easy to evaluate for both users and developers.

# Changes to the Requirements:

Over the course of development, several notable changes were made to the feature set and overall scope of the application rather than the system requirements themselves. While the technical requirements (such as minimum OS, RAM, and CPU) remained consistent, the team made changes to the functional and usability aspects based on user testing, development time, and feasibility.

**Removed Features:**

Online Integration (such as user accounts or syncing) was scrapped due to time limitations and to keep the app offline-first and lightweight.

Advanced calculator modes that required complex math libraries were delayed for future and public versions.

**Added Features:**

Settings Options such as theme toggling (dark/light mode), tile resizing, and a visibility toggle to hide unused calculators were implemented to increase user customization.

A filter sidebar was introduced to improve calculator discoverability based on categories or functions.

Ease of access settings like increased contrast or text size support are being planned for inclusivity and accessibility.

**Revised Usability Priorities:**

Inspired by the 10 Usability Heuristics, the prototype was adjusted to focus more on minimalism, consistency, recognition over recall, and user freedom. For instance, instead of overwhelming the user with all calculators at once, users can now filter or hide tools they don’t use.

These adjustments reflect the team’s focus on improving user experience while staying within scope and time constraints.

# Initial Evaluation Plan:

With the application now fully buildable in Android Studio and testable via emulator or APK, the team is conducting a structured evaluation of the prototype through a combination of in-device testing, user-centered observation, and heuristic inspection. The primary goal is to assess usability, performance, and user satisfaction prior to final deployment.

The evaluation process consists of the following components:

1. Functional Testing on Emulators and Real Devices The team conducts technical testing using Android Studio’s emulator (API Level 36, Medium Phone) and actual Android smartphones. This ensures functionality across different screen sizes, resolutions, and Android versions. Testing focuses on:

* Calculator accuracy
* UI responsiveness
* Filter/search functionality
* Settings interactions

1. User Testing and Observation A task-based usability test is administered with ~10 participants using a test device. Participants are observed live and optionally recorded (with consent). This technique evaluates:

* How effectively users complete real-world tasks
* What errors or frustrations arise
* How long tasks take to complete
* What design changes are needed

Participants are not guided, but they are encouraged to think aloud while using the prototype. Their behavior and comments will be noted to gain qualitative insight into usability challenges.

1. Bug and Usability Issue Tracking All functional bugs and usability-related observations are logged in a shared issue tracker (e.g., GitHub Issues). Each issue is categorized by:

* Severity (critical, moderate, minor)
* Frequency
* Usability impact

**Usability Specifications**

The creation of this prototype will aim to achieve the following measures when it appeals to the use:

* **Effectiveness**: When accomplishing this measurement, it will show how well the prototype is at performing the required tasks.
* **Efficiency**: This measurement aims to show how easy and simple the prototype is used.
* **Utility**: This aims to show that the prototype will support suitable functions and alternatives to certain tasks
* **Learnability**: This will showcase how easy the users will learn to use the prototype system.
* **Memorization**: This will showcase how simple the users can remember steps when using the system.

# Target Population

10 participants will engage with the prototype using defined task flows to simulate real user behavior. They will be expected to complete tasks independently while being observed and later surveyed.

# Prototype Tasks

The tasks for this Prototype are split into three (3) different Sections: Exploration Tasks, Calculator Tasks, and Settings Tasks. Below are some of the tasks that the selected participants will be asked to perform for each Section to showcase the Prototype’s functionality:

* How easy will the user be able to navigate while using the Prototype.
* Participants will be tasked in exploring the application (**Exploration Tasks**)
* Participants will be tasked in interacting with any of the calculators (**Calculator Tasks**)
* Participants will be tasked in interacting with the Settings (**Calculator Tasks**)
* Participants will be tasked to edit settings with their preferences (**Settings Tasks**)

Reasons that these tasks were selected for the participants since the Prototype was designed with these measures in mind:

* Easy Navigation
* Core Functional Use
* Customization and Control

|  |  |  |  |
| --- | --- | --- | --- |
| Exploration | Within 2 minutes or Below | Highly Acceptable | Successful |
| Above 2 minutes | Not Acceptable | Unsuccessful |
| Calculators | Within 5 minutes or Below | Highly Acceptable | Successful |
| Above 5 minutes | Not Acceptable | Unsuccessful |
| Settings | Within 5 minutes or Below | Highly Acceptable | Successful |
| Above 5 minutes | Not Acceptable | Unsuccessful |

Table 2. Time Interpretation

# Heuristic Evaluation

Evaluation of CalcBox will also use the 10 Usability Heuristic method of Evaluation.

*Visibility of System Status*

CalcBox keeps users informed about their actions status through real-time calculator outputs, and toggle feedback. Tapping a tile shows a clear transition to the selected calculator, so users always know what the app is doing.

*Match Between System and Real World*

The app uses simple, familiar language such as “BMI Calculator,” “Currency Converter,” or “Hide Calculator” rather than technical or system terms. Calculators are grouped by recognizable categories, making the layout intuitive even for casual users.

*User control and Freedom*

CalcBox allows users to hide calculators they don't need and restore them from settings without permanently deleting them. They can easily return to the main screen from any point in the app, avoiding unwanted states or confusing navigation flows.

*Consistency and Standards*

Icon usage, button placement, and navigation patterns remain consistent throughout the app. Actions such as going back, opening a calculator, or modifying settings behave the same across all screens, reducing confusion.

*Error Prevention*

Input fields in calculators validate entries such as preventing letters in numeric fields, reducing the chance of user error. Where applicable, guidance text helps users enter data correctly before submitting.

*Recognition rather than recall*

CalcBox uses icons, labels, and tooltips to guide users instead of relying on memory. Users can scroll, search, or filter through visible tiles instead of having to remember calculator names or functions.

*Flexibility and Efficiency of Use*

The app is designed for both novice and advanced users. New users can scroll through and explore visually, while experienced users can jump directly to calculators using the search or filters. Users can also personalize the layout for quicker access to their most-used tools.

*Aesthetic and Minimalist Design*

CalcBox keeps its interface clean and focused only showing essential calculator tiles, basic settings, and user-friendly icons. Non-critical information is hidden or tucked away in settings, keeping the main interface distraction-free.

*Help Users Recognize, Diagnose, and Recover from Errors*

When errors occur (e.g., invalid input), the app provides helpful messages like “Please enter a valid number” instead of vague or coded alerts. These are displayed near the input fields to help users quickly fix their mistakes.

*Help and Documentation*

A Help or FAQ section is accessible through the Settings page. This section explains how to use key features such as customizing the tile layout, hiding calculators, and resetting preferences. It’s concise and easy to navigate when assistance is needed.

The evaluation plan ensures that CalcBox is not only functional but user-centered, easy to navigate, and customizable. Each technique plays a role in improving the prototype:

* Heuristic Evaluation for expert-driven insight
* User Testing for real-world usability observation
* Surveys for quantifying satisfaction and ease of use

These combined approaches help refine the app before deployment and ensure its usability and success among target users.