Project Proposal

# Project Overview

## Project Title

Hand Control App

## Client Name

## Project Summary

Develop an Android application that uses the front-facing camera and on-device ML to recognize hand gestures and translate them into system commands, enabling true touch‑free control of the smartphone for accessibility and convenience.

## Objectives

* Replicate the hand control functionality found in Huawei and Honor devices
* Enable users to control smartphone actions using gestures without touching the screen
* Provide a smart activation mechanism that works only when the user is looking at the screen
* Deliver a battery‑efficient, low‑latency solution

# Business Requirements

## Goals

* Build a functional Android app with accurate real‑time hand gesture recognition
* Achieve high detection accuracy while minimizing battery impact
* Launch the app targeting accessibility and hands‑free utility markets
* Establish a foundation for future monetization or licensing opportunities

## Target Audience

Smartphone users seeking hands‑free interaction, especially individuals with accessibility needs

## Expected Outcomes

* A stable Android app that can scroll, swipe, capture screenshots, and activate via gestures
* Positive user feedback on ease of use and reliability
* Compliance with Google Play policies regarding camera and accessibility services

## Success Metrics

* Gesture recognition accuracy ≥ 90%
* Battery usage increase ≤ 5% during typical usage
* User satisfaction score ≥ 4 out of 5 in beta testing
* Number of downloads or installations within the first 3 months

## Monetization Strategy

# Technical Requirements

## Core Features

* Scroll via hand wave up/down
* Swipe left/right with palm swipe
* Screenshot capture with fist clench
* Smart activation based on face detection

## Core Features & Functionalities

* Real‑time hand landmark detection using Google ML Kit pose detection
* Camera feed handling via CameraX
* Mapping of gesture patterns to system actions using Android Accessibility Service
* Face detection to enable smart activation only when user looks at screen
* Adjustable sensitivity and battery‑optimization settings

## Tech Stack

* Android (Kotlin)
* CameraX
* Google ML Kit (on‑device pose detection)
* Android Accessibility Service
* Android Jetpack components

## Integration Needs

## Security & Compliance

* Request and handle CAMERA permission transparently
* All video processing performed on device, no external data transmission
* Compliance with Google Play privacy policy and accessibility service guidelines

## Performance Criteria

* Detection latency < 100 ms per frame
* Memory footprint ≤ 100 MB
* Battery impact minimal, with smart activation reducing unnecessary processing

# App Flow

## App Flow Summary

* User launches the app
* App requests camera permission
* Live camera feed starts via CameraX
* Face detection checks if user is looking at screen (smart activation)
* ML Kit processes frames to identify hand landmarks
* Recognized gesture is matched to predefined actions
* Accessibility Service executes the corresponding system command
* Feedback (visual/toast) is shown to confirm the action

# Project Scope

## Inclusions

* Design and development of the Android Hand Control app
* Implementation of scroll, swipe, screenshot, and smart activation features
* Integration with Android Accessibility Service
* Full testing cycle including beta testing
* User documentation and developer hand‑over

## Exclusions

* iOS version development
* Custom deep‑learning model training
* Backend server components or cloud services

## Deliverables

* Complete source code repository
* Signed release APK (or AAB) ready for Google Play
* Technical design document
* Test plan and test results
* User guide and quick‑start tutorial

## Milestones

* Phase 1 – Proof of Concept: Camera feed and ML Kit hand detection
* Phase 2 – Gesture Logic & Actions: Map gestures to UI feedback
* Phase 3 – System Integration: Accessibility Service and smart activation
* Phase 4 – Polishing & Testing: Optimization, beta testing, bug fixes
* Final Release: Production build and deployment preparation

## Estimated Timeline & Pricing

# Timeline & Resources

## Estimated Duration

## Team Roles

* Project Manager
* Android Developer
* Machine Learning Engineer
* UI/UX Designer
* Quality Assurance Tester

## Dependencies

* Google ML Kit SDK
* CameraX library
* Android OS version 8.0 (API 26) or higher
* Device camera hardware

# Budget & Costing

## Estimated Budget

## Cost Breakdown

No cost breakdown provided.

# Risk Assessment

## Potential Risks

* Inaccurate gesture recognition leading to false actions
* High battery consumption if smart activation fails
* Privacy concerns around camera usage
* App rejection due to Accessibility Service policy violations

## Mitigation Strategies

* Implement extensive gesture testing and adjustable sensitivity thresholds
* Optimize processing pipelines and enforce smart activation checks
* Ensure all video data stays on device and provide clear permission dialogs
* Conduct compliance review with Google Play policies before submission