
BlueCat

— Interactive Productivity Manager —
Team 2

Rachel Gan, Young Woong Jun, Sigrid Marita Kvamme, Jin Yeong Kim, Seung Woo Jo

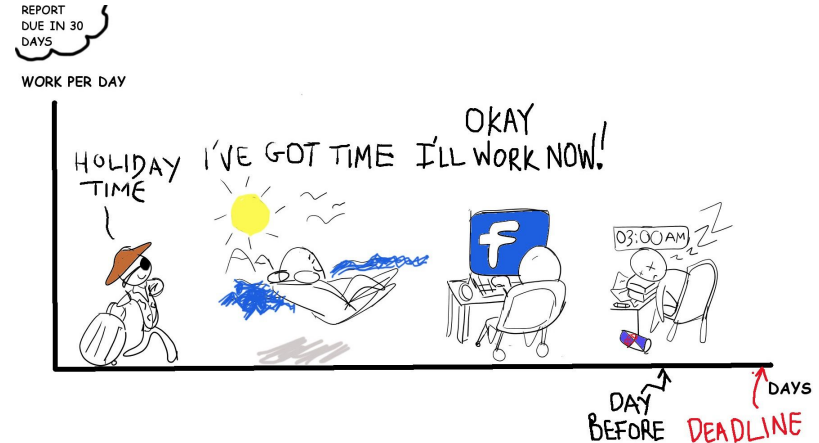
Project Idea Interactive Productivity Manager

Target Audience

- Students or workers who have problems meeting deadlines
- Procrastinators and people who get distracted easily

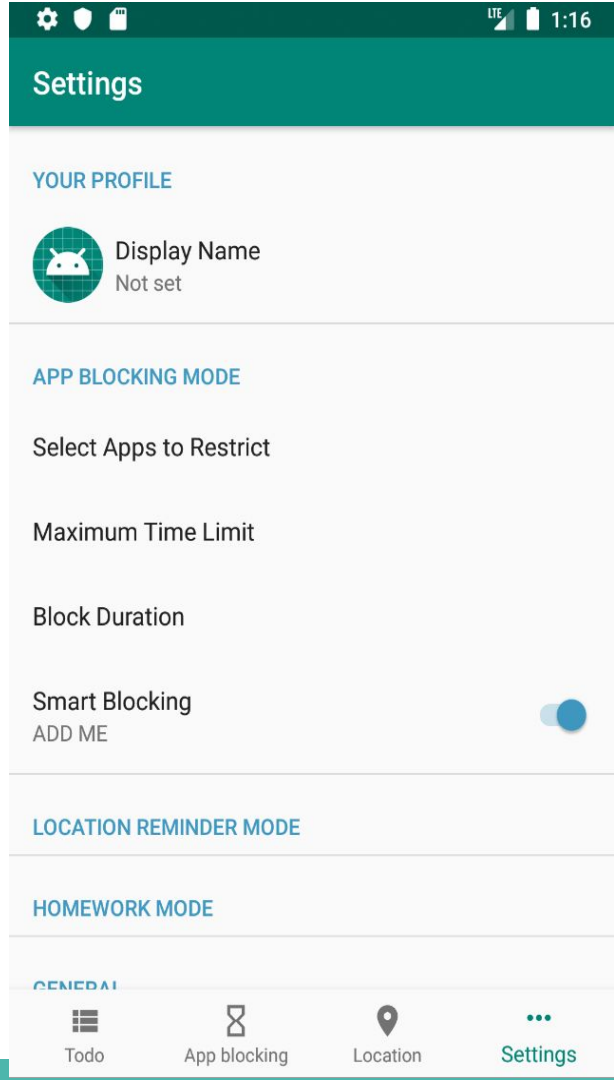
Goal

- Prevent users from procrastinating and help them to increase their productivity while staying happy



Quick Architecture

- Three separate features
 - To-do List & HW Mode / Location-Reminder / App-Blocking
- Common settings page
 - Controls settings for each function at one page
- Local Database using Room
 - Separate database for each function
- Foreground service
 - App-blocking features
 - Location-reminder features



Project Schedule

Spreadsheet (Project Plan , Check up on the progress) :

<https://docs.google.com/spreadsheets/d/1QyvoBminOBTyXSOlePHeygPNn8a5bCXSTumpKnIcDBE/edit#gid=1533464390>

Git Issues (Easy Daily Management) : <https://github.com/ywj7373/MC-group2/issues>

What has been done

- Getting familiar with Android development with Kotlin
- Implement basic functionality of the project (details later)

What will be done

- Implement the features that have not been completed yet (details later)

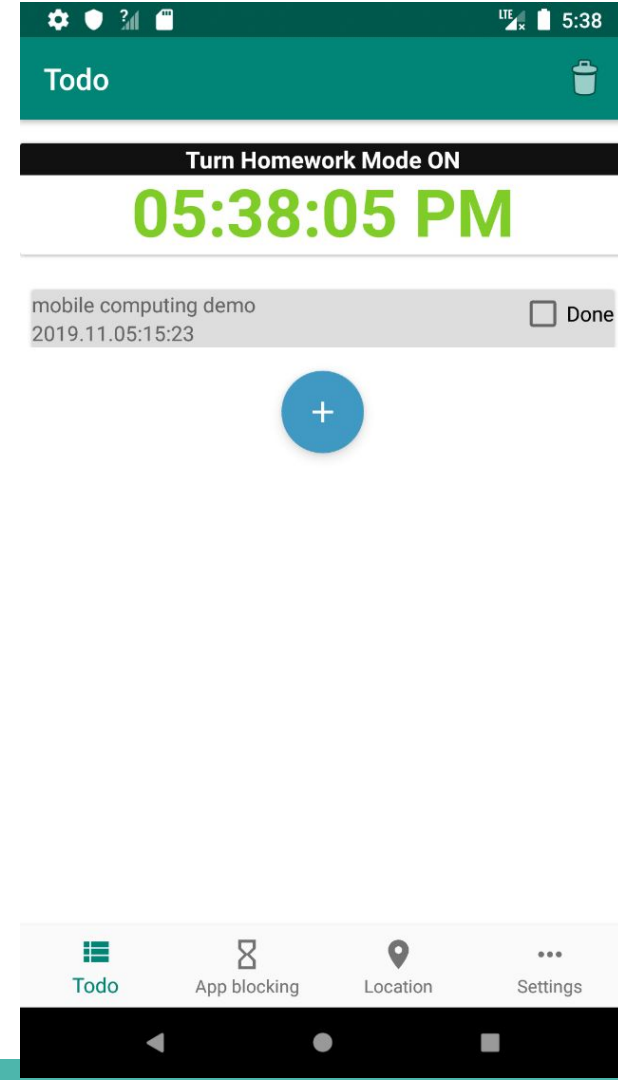
Project Schedule - Deviations

- **To-do List & Homework mode**
 - Have not made basic version of motion detector
- **App blocking**
 - Step count feature for unblocking not fulfilled (not a core feature)
 - Not implemented information notifications
- **Location-Reminder**
 - Notification alert not yet implemented
 - Not implemented gamification

To-Do List & HW Feature

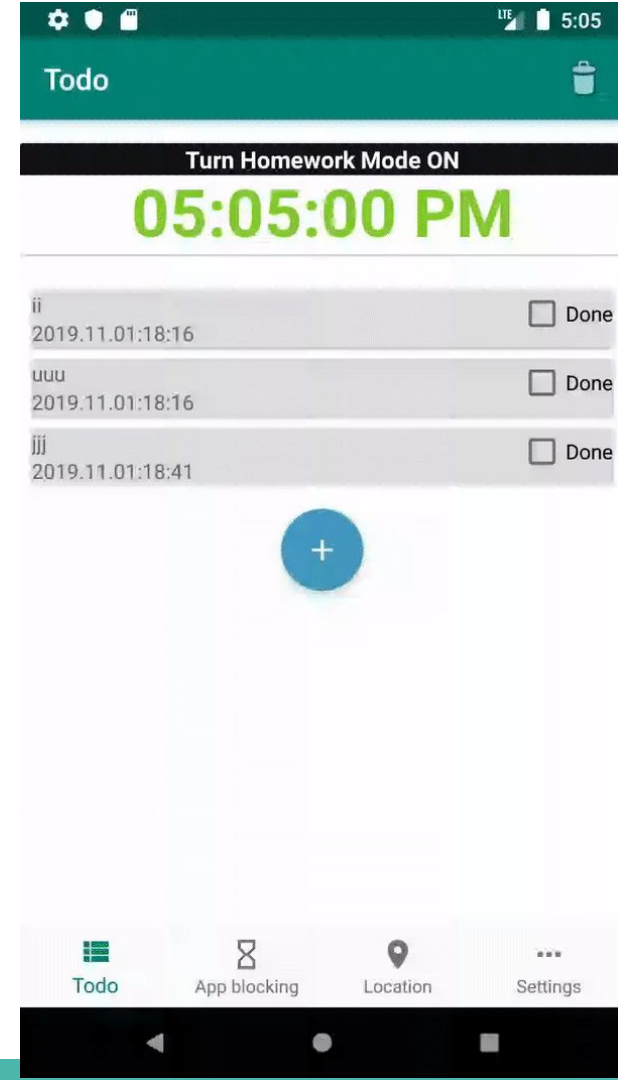
To-Do List & Homework Mode DEMO

- **Common To-Do List functions**
 - Mark tasks done, undone
 - add, delete tasks
- **Homework Mode**
 - Homework mode activates instant app blocking on the list of apps set in the settings page
 - Alarm goes off if a user does not check in before timer ends
 - Alarm can be turned off by making some motions (shaking the phone 10 times or walk 30 steps around)



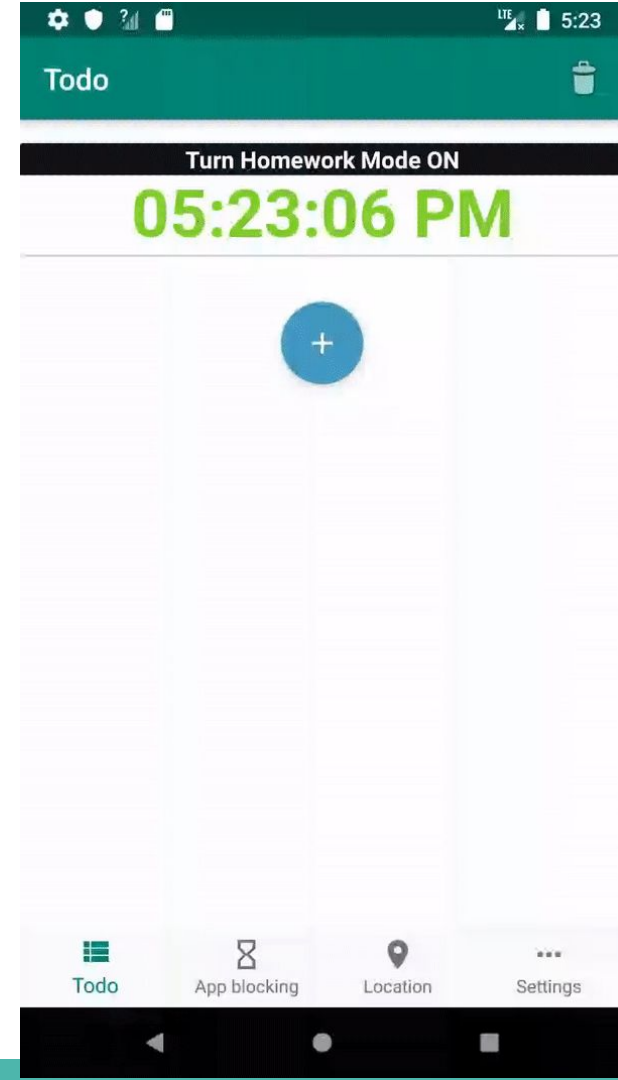
To-Do List & Homework Mode DEMO

- **Common To-Do List functions**
 - **Mark tasks done, undone**
 - **add, delete tasks**
- Homework Mode
 - Homework mode activates instant app blocking on the list of apps set in the settings page
 - Alarm goes off if a user does not check in before timer ends
 - Alarm can be turned off by making some motions (shaking the phone 10 times or walk 30 steps around)



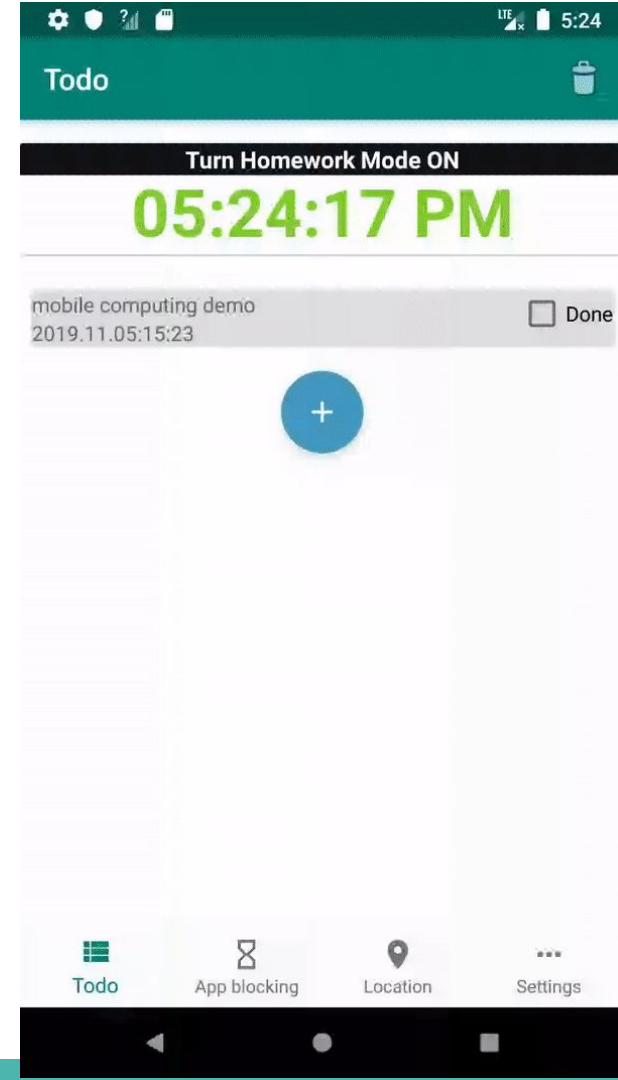
To-Do List & Homework Mode DEMO

- **Common To-Do List functions**
 - **Mark tasks done, undone**
 - **add, delete tasks**
- Homework Mode
 - Homework mode activates instant app blocking on the list of apps set in the settings page
 - Alarm goes off if a user does not check in before timer ends
 - Alarm can be turned off by making some motions (shaking the phone 10 times or walk 30 steps around)



To-Do List & Homework Mode DEMO

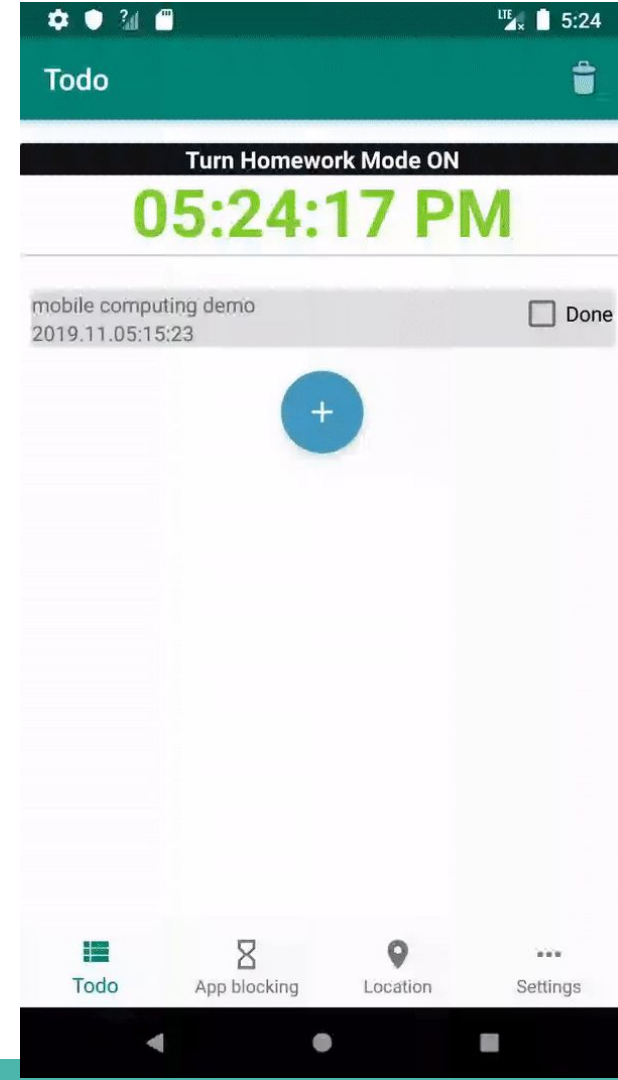
- Common To-Do List functions
 - Mark tasks done, undone
 - add, delete tasks
- **Homework Mode**
 - Homework mode activates instant app blocking on the list of apps set in the settings page
 - Alarm goes off if a user does not check in before timer ends
 - Alarm can be turned off by making some motions (shaking the phone 10 times or walk 30 steps around)



To-Do List & Homework Mode DEMO

Scenario

- Must finish Mobile demo presentation slides by tomorrow
- But, a new mobile game I've started playing is so fun.
- So, I block the game (add it to the blocked app list) and turn on Homework Mode
- I can't play the game until I turn off HW mode
- When I try to turn off HW mode even though I haven't finished my homework, I feel guilty because of the text that I have to type in.
- So, I keep working on the slides and finish them and happily play the game later :)
- If I fall asleep while doing the homework, the alarm wakes me up.



Going forward To-Do List & Homework Mode

Things to be implemented

- Stable Timer
- Send Notification 5 minutes prior to timer ends (with vibration)
- Make alarm go off as timer ends
- Use motion data to turn off the alarm

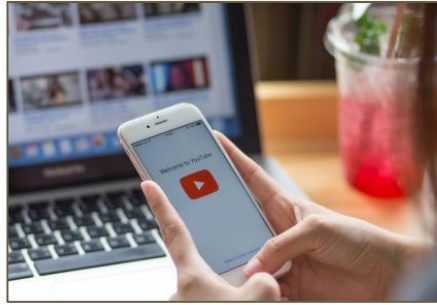
Expected technical challenges

- Keeping the timer going as the app goes to the background
- Detecting user motions

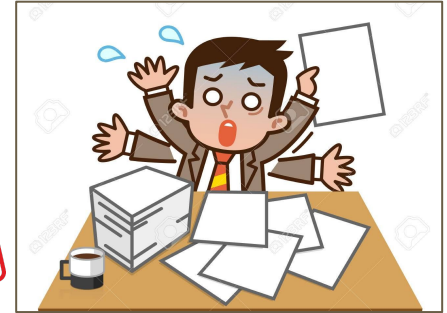
App-Blocking Feature

App Blocking Use Case

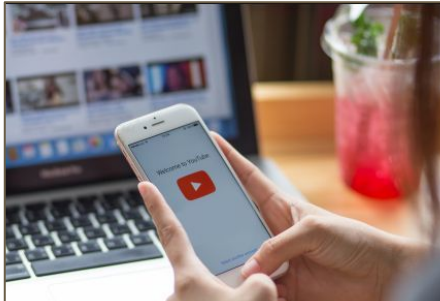
Without BlueCat



**LAST
MINUTE**



With BlueCat

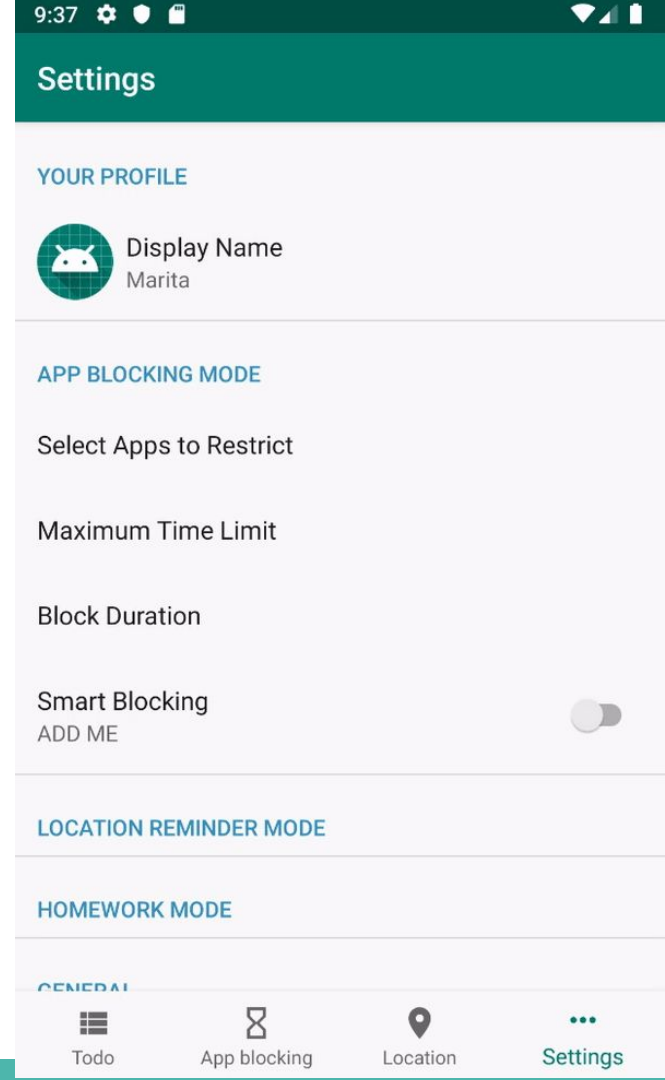


BLOCKED



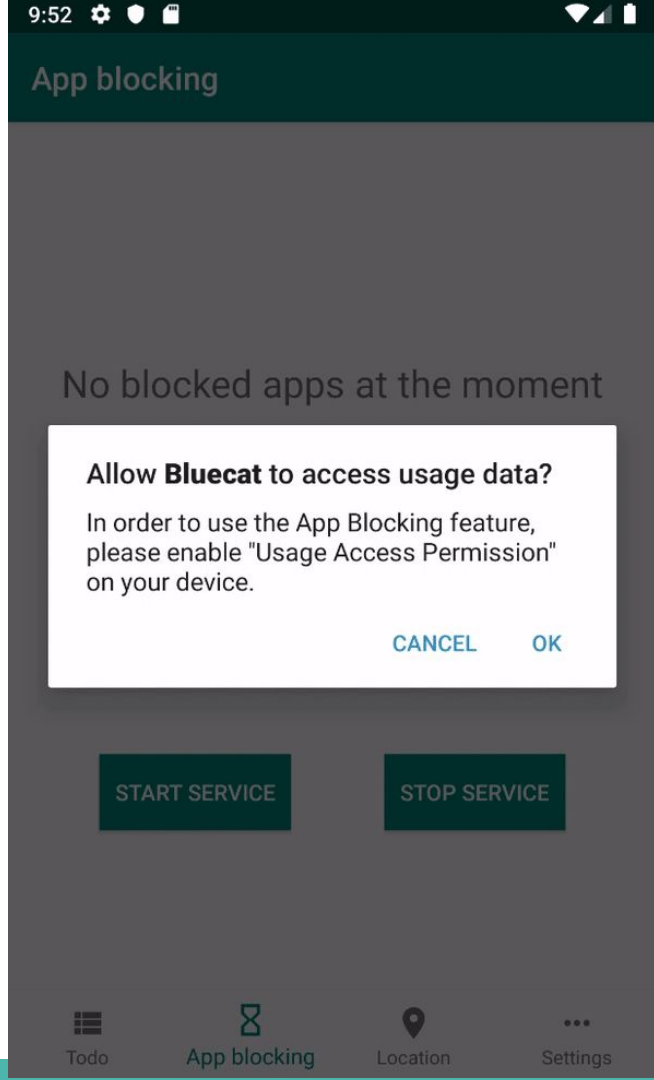
App Blocking DEMO

- **User chooses custom preferences**
 - Select apps to block from app list
 - Choose time limit for app usage (default 30min)
 - Choose block duration (default 5mins)
- Usage access permissions
 - Requests for usage access permission
- App blocking
 - If you use a restricted app for more than the limit, it will block the app and redirect you to block countdown view
 - Resets usage timer if you take a break



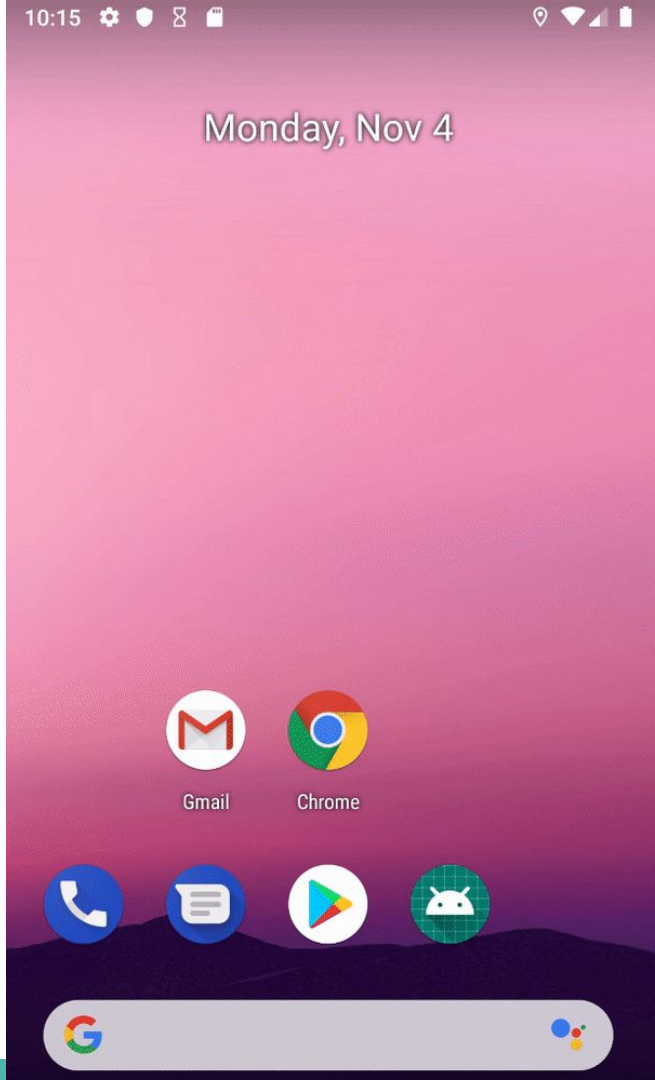
App Blocking DEMO

- User chooses custom preferences
 - Select apps to block from app list
 - Choose time limit for app usage (default 10s)
 - Choose block duration (default 5mins)
- **Usage access permissions**
 - **Requests for usage access permission**
- App blocking
 - If you use a restricted app for more than the limit, it will block the app and redirect you to block countdown view
 - Resets usage timer if you take a break




App Blocking DEMO

- User chooses custom preferences
 - Select apps to block from app list
 - Choose time limit for app usage (default 10s)
 - Choose block duration (default 5mins)
- Usage access permissions
 - Requests for usage access permission
- **App blocking**
 - **If you use a restricted app for more than the limit, it will block the app and redirect you to block countdown view**
 - **Resets usage timer if you take a break**



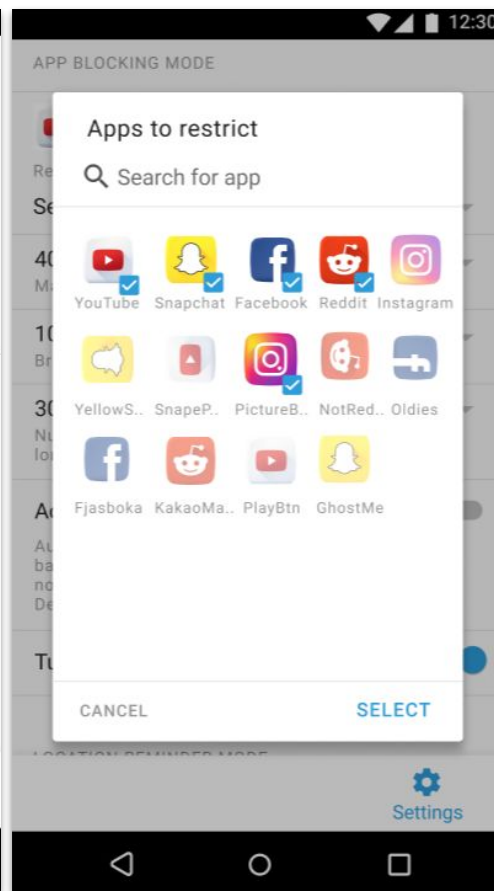
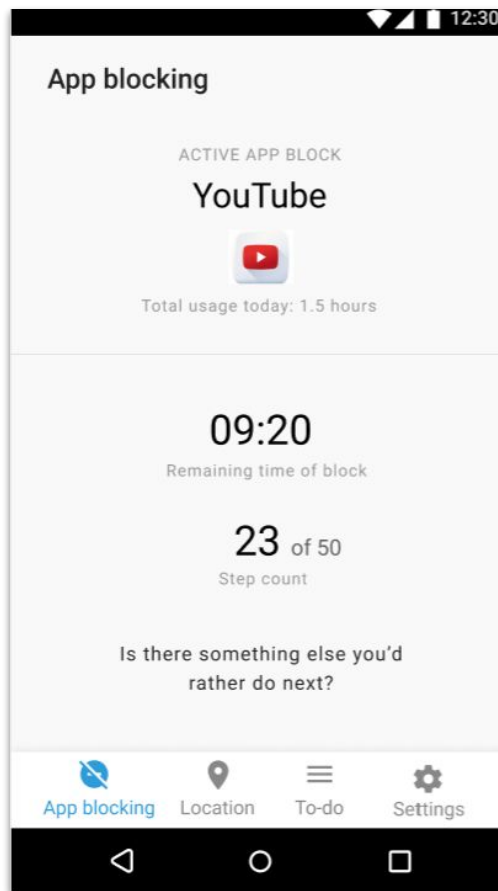
App Blocking Key Technical Challenges

Challenges		Solutions
Block app based on app usage in a smart way.		Develop an algorithm based on different types of data from device to guess when user is procrastinating (yet to be developed)
Unblock the app after a set amount of time		Remove app from blocking list after internal countdown timer finishes
Blocking multiple apps (each app needs its own countdown timer)		Store a map containing the blocks for each app, display countdowns of all blocked apps

Going forward App Blocking

Things to be implemented:

- **Block screen for multiple apps**
- **Notifications to inform user**
- **Pedometer feature**
 - Prompt users to walk a number of steps in order to unblock app
- **Smart blocking feature**
 - Detects procrastination
 - Automatically blocks apps if you have items on todo-list (with warning)
- **Setting to disable feature**



Location-Reminder Feature

Location-Reminder Use-case

Without BlueCat



**LAST
MINUTE**

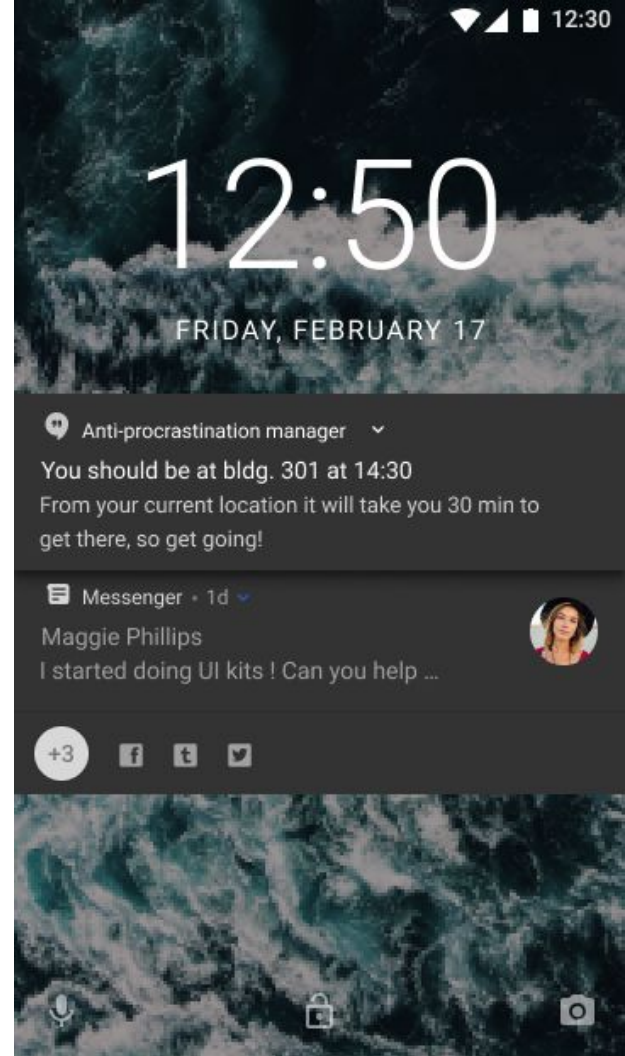


With BlueCat



Location-Reminder overview

- The app will set off an alarm based on your schedule
- It will consider your current location, and the travel time to your scheduled destination, to calculate when to set off the alarm
- **For example, if you take 1 hour to reach your scheduled location, the app will alert you 1 hour + alpha beforehand!!**
- No need to manually set an alarm every time!



Location-Reminder Demo

- Adding Item to Schedule List
 - Can search location ← Used Naver Map API
 - Select either date or days
 - Save and Keep schedule ← Used Room DB
- Check every one minute
 - Call routine task every minute ← Used Receiver, Service(Foreground)
 - Get current position ← Used FusedLocationProviderClient
 - Estimate travel time ← Used ODSay API



Location-Reminder Demo

- **Adding Item to Schedule List**
 - Can search location ← Used Naver Map API
 - Select either date or days
 - Save and Keep schedule ← Used Room DB
- Check every one minute
 - Call routine task every minute ← Used Receiver, Service(Foreground)
 - Get current position ← Used FusedLocationProviderClient
 - Estimate travel time ← Used ODSay API

10:13 LTE

BlueCatApp


Location Schedule

서울역 경부선(고속철도)

Select time

10	14	
11	15	AM
12	16	PM

Select date or days

2019-11-04 

SUN	MON	TUE	WED	THU	FRI	SAT
-----	-----	-----	-----	-----	-----	-----

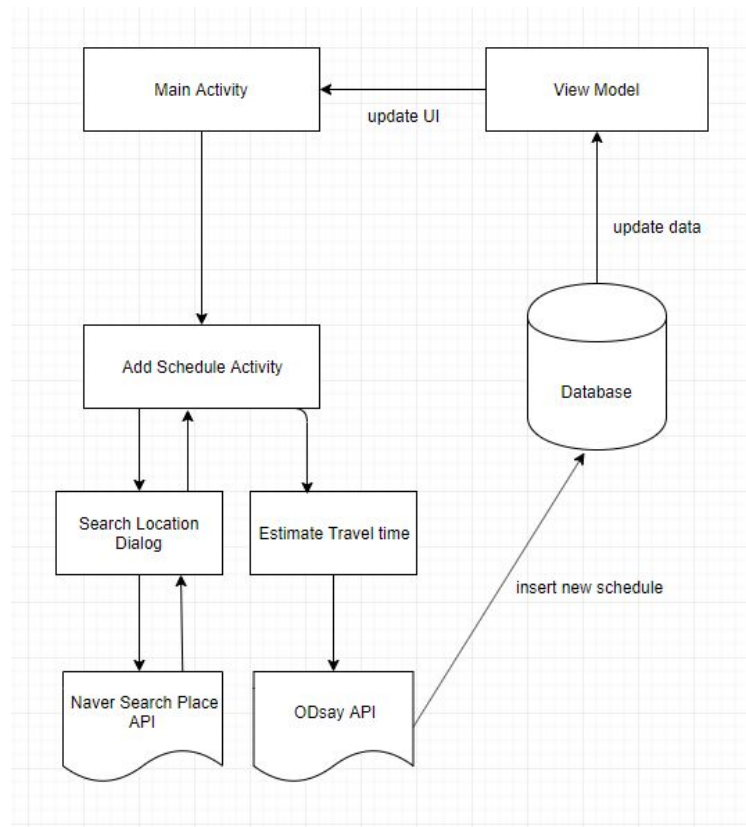
Location-Reminder Demo

- Adding Item to Schedule List
 - Can search location ← Used Naver Map API
 - Select either date or days
 - Save and Keep schedule ← Used Room DB
- **Check every one minute**
 - **Call routine task every minute** ← Used Receiver, Service(Foreground)
 - **Get current position** ← Used FusedLocationProviderClient
 - **Estimate travel time** ← Used ODSay API

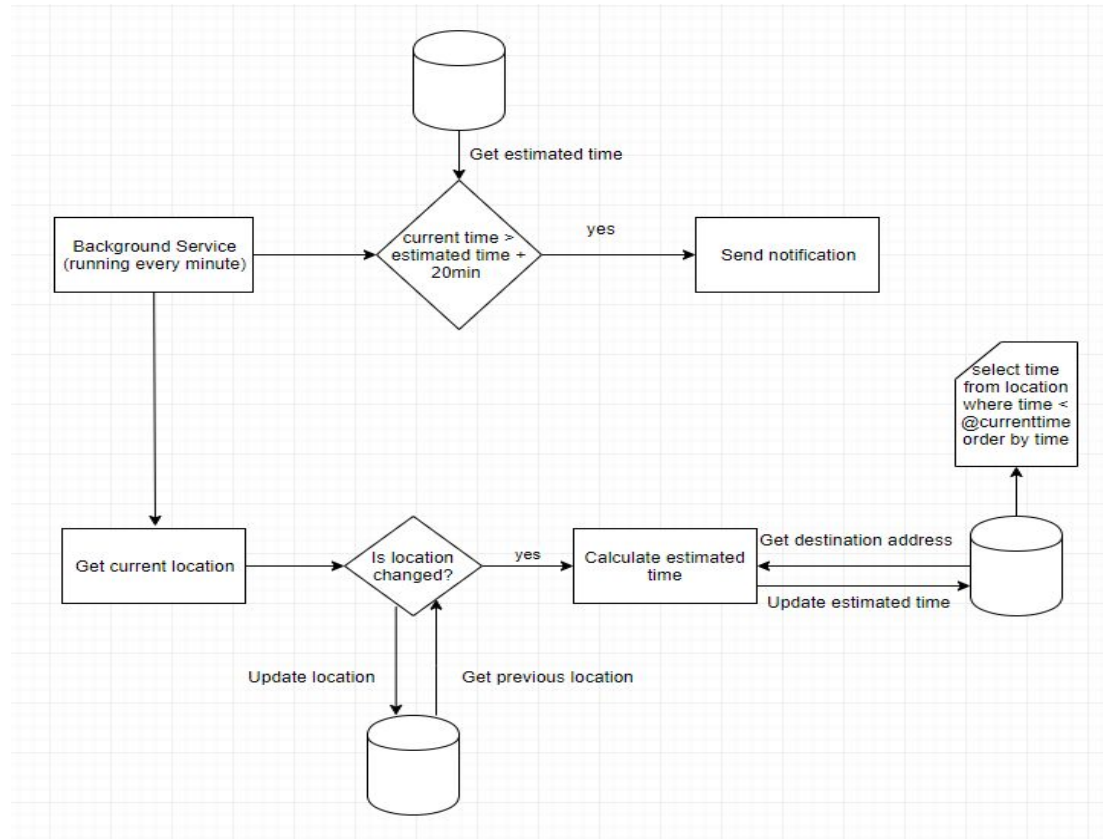
Location-Reminder Demo

```
2019-11-04 19:56:54.335 24440-24440/com.example.bluecatapp D/RoutineService: 0.0
2019-11-04 19:57:56.301 24440-24440/com.example.bluecatapp D/RoutineService: 0.0
2019-11-04 19:58:57.316 24440-24440/com.example.bluecatapp D/RoutineService: 2.0728724024724468
2019-11-04 19:58:57.319 24440-24924/com.example.bluecatapp D/RoutineService: Current Location Updated
2019-11-04 19:58:57.321 24440-24924/com.example.bluecatapp D/RoutineService: 서울대학교관악캠퍼스제1공학관
2019-11-04 19:58:58.142 24440-24440/com.example.bluecatapp D/RoutineService: Connection to Odsay successful
2019-11-04 19:58:58.149 24440-24440/com.example.bluecatapp D/RoutineService: Estimated Time of 서울대학교관악캠퍼스제1공학관 changed to 23
2019-11-04 20:00:00.888 24440-24440/com.example.bluecatapp D/RoutineService: 0.0
2019-11-04 20:01:01.912 24440-24440/com.example.bluecatapp D/RoutineService: 2.4788780872873755
2019-11-04 20:01:01.913 24440-24945/com.example.bluecatapp D/RoutineService: Current Location Updated
2019-11-04 20:01:01.923 24440-24945/com.example.bluecatapp D/RoutineService: 서울대학교관악캠퍼스제1공학관
2019-11-04 20:01:02.599 24440-24440/com.example.bluecatapp D/RoutineService: Connection to Odsay successful
2019-11-04 20:01:02.600 24440-24440/com.example.bluecatapp D/RoutineService: Estimated Time of 서울대학교관악캠퍼스제1공학관 changed to 28
2019-11-04 20:02:02.929 24440-24440/com.example.bluecatapp D/RoutineService: 0.0
```

Location-Reminder Activity Architecture



Location-Reminder Service Architecture



Location-Reminder Key technical challenges

Power consumption

- **Used FusedLocationProviderClient** ← **Better battery usage**
- **Used inexact 1 minute check interval** ← **Synchronize with other services**
- **Minimize redundant operation**

Android prohibit background location tracking

- Used ForegroundService
- Used FusedLocationProviderClient

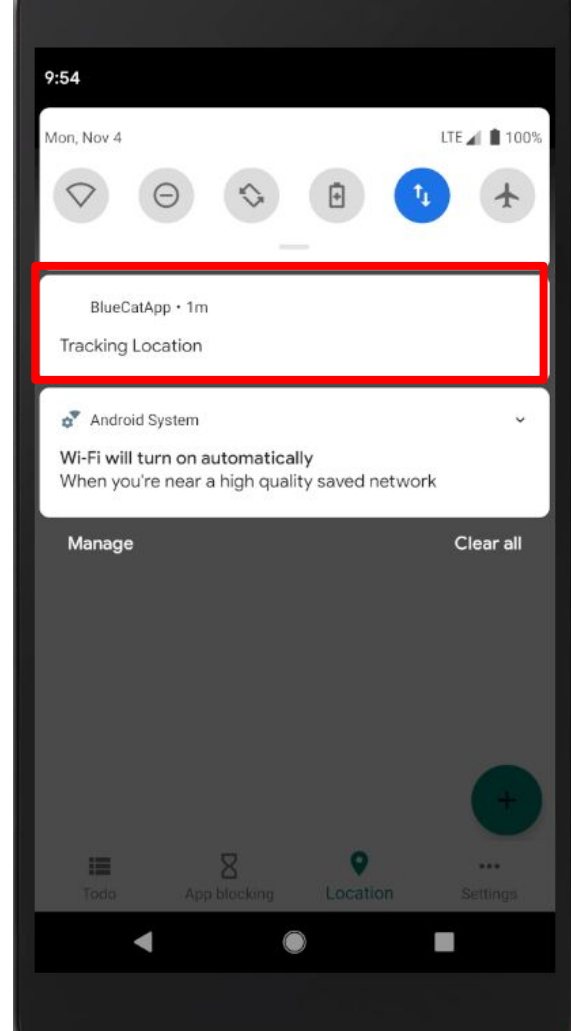
Location-Reminder Key technical challenges

Power consumption

- Used FusedLocationProviderClient ← Better battery usage
- Used inexact 1 minute check interval ← Synchronize with other services
- Minimize redundant operation

Android prohibit background location tracking

- **Used ForegroundService**
- **Used FusedLocationProviderClient**



Going forward Location-Reminder

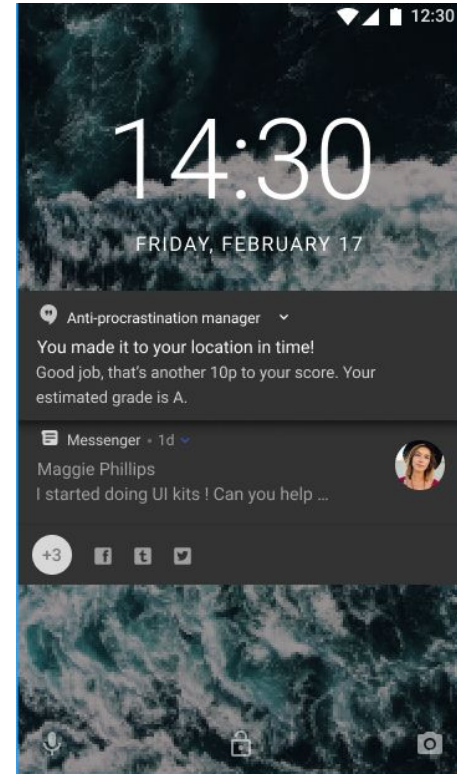
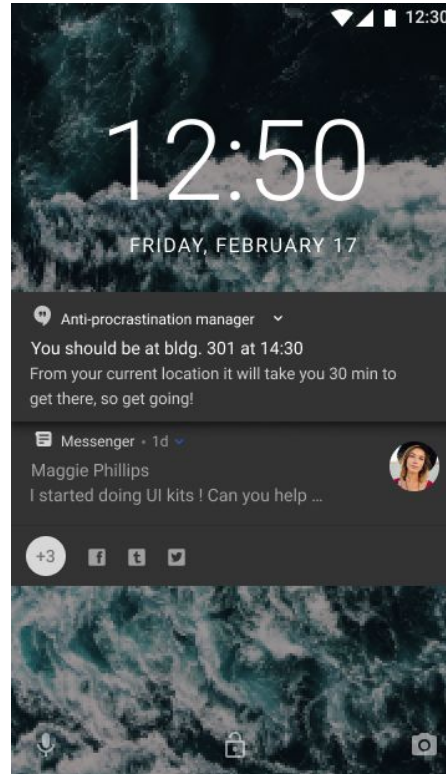
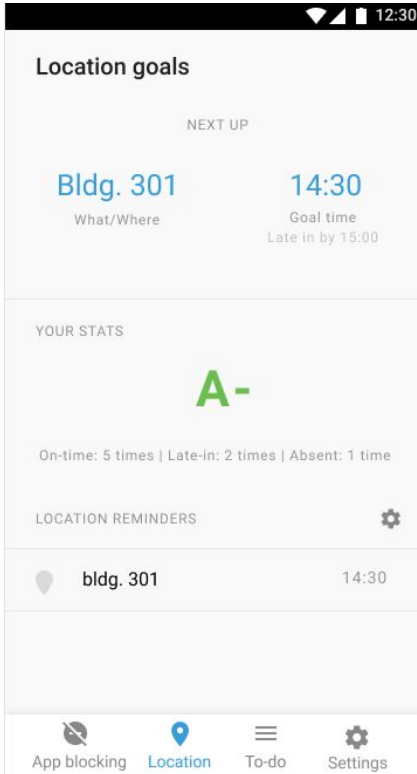
Not yet implemented

- Gamification
- Alarm or Notification*
- Multiple test cases
- Various settings (turn on/off, duration, result in english?)

Things to improve

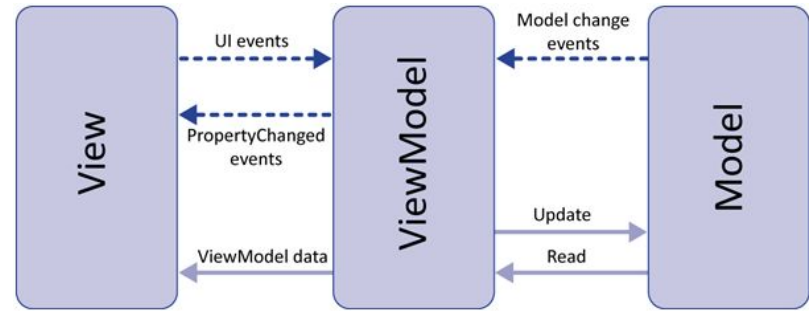
- UI
- Better ways to decrease power consumption

Location-Reminder Final Product



Key Takeaways from first part of project

- Intro to Android Programming with Kotlin
- Android MVVM pattern
- Usage of different APIs
- Implementation of foreground services to keep app functionality alive
- There is a lot of restrictions and APIs in Android that makes developing challenging!



Final Deliverable & Success Criteria

Todo List & Homework Mode

- Stable homework mode that keeps track of time while the mode is on.
- Send notification and fire the alarm at proper time.
- Accurate detection of user motions (shaking and walking)

App Blocking

- Successfully block selected apps based on settings configuration
- Unblock app after step count completed and/or timer completed

Location-Reminder

- Stable location tracking and travel time estimation
- Send notification and fire the alarm at proper time.
- Reward and penalize properly in gamification

