

CSCI4100U: Mobile Devices  
Project Proposal

Syed Naqvi 100590852  
Brendan Murray 100825624  
Evan Goldenberg 100826897

October 13, 2024

# Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Core Features</b>	<b>3</b>
<b>3</b>	<b>Technical Specifications</b>	<b>3</b>
<b>4</b>	<b>Work Distribution</b>	<b>4</b>
<b>5</b>	<b>Application Design</b>	<b>4</b>
<b>6</b>	<b>Mockup User Interface</b>	<b>5</b>

# 1 Overview

The purpose of this document is to propose the development of a mobile application titled **Task Bell**. This app offers an effective solution for shift-workers and individuals requiring multitasking or productivity management assistance.

## 2 Core Features

This section outlines the key features and functionalities of the proposed mobile application.

- **Alarm Grouping:**  
The ability to create a hierarchical alarm grouping structure, similar to a directory tree, that can be used to set multiple alarms at once. Settings of a parent group can be applied locally or propagated to all child groups. Individual groups or alarms can optionally be set to "inheritance blocking mode" making them immune from any propagated settings.
- **Date/Time - Based Recurrence Settings:**  
The user will be able to set recurrence patterns for their alarms/groups based on calendar dates and times of the day. There will also be the inclusion of a relative adjustment feature that can shift all times based on the time of the first alarm/timer of the group.
- **Location - Based Recurrence Settings:**  
The user can choose to make their alarms/groups recur based on their location.
- **Alarm Disabling Tasks:**  
The alarms/groups can be set so that tasks such as object/QR code scanning or puzzle completion are required before any particular alarm can be disabled.
- **Timer/Stopwatch Support:**  
The app will allow standard stopwatch and timer support including multiple stopwatches and/or timers running simultaneously.
- **Basic Functional Requirements:**  
The user's saved alarms/groups as well as the associated recurrence settings (both data/time and location based) will require both local and cloud databases for storage and device synchronization. Location based recurrence will involve geolocation and http requests. The app will contain multiple screens for stopwatches, alarms and timers while utilizing dialogs/pickers and snack bars for interacting with and modifying alarms. Notifications will be heavily leveraged for upcoming alarms and active alarms.

## 3 Technical Specifications

The app will be built using the following technologies:

- **Programming Languages:** Dart
- **Frameworks:** Flutter
- **Platforms:** Android platform
- **Database:** SQLite
- **APIs:** Fused Location Provider API, Google Maps Android API, Geofencing API, Dart:Core - DateTime API

## 4 Work Distribution

The planned workload distribution is as follows:

Team Member	Task
Syed Naqvi	<ul style="list-style-type: none"> <li>- Back-end for location-based recurrence and disabling tasks</li> <li>- Front-end/UI design</li> </ul>
Evan Goldenberg	<ul style="list-style-type: none"> <li>- Back-end for timer/stopwatch and date/time recurrence</li> <li>- Date/time UI as applies</li> </ul>
Brendan Murray	<ul style="list-style-type: none"> <li>- Back-end for alarm grouping and date/time recurrence</li> <li>- Date/time UI as applies</li> </ul>

## 5 Application Design

The following UML diagram is intended to provide an overview of application usage:

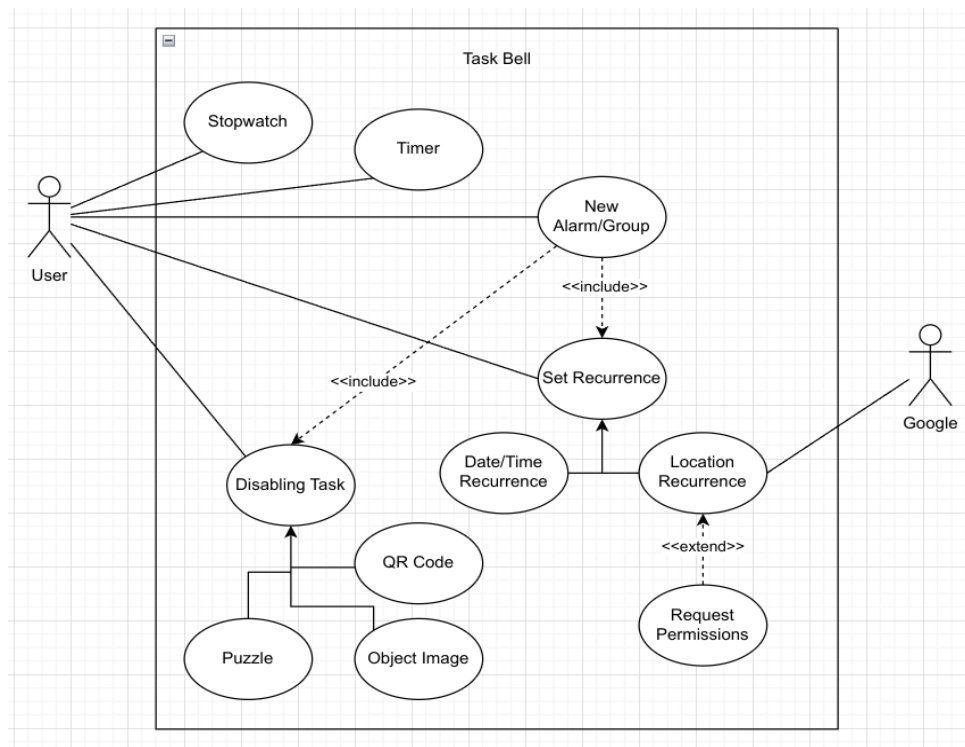


Figure 1: Use Case Diagram

## 6 Mockup User Interface

Initial user-interface design:

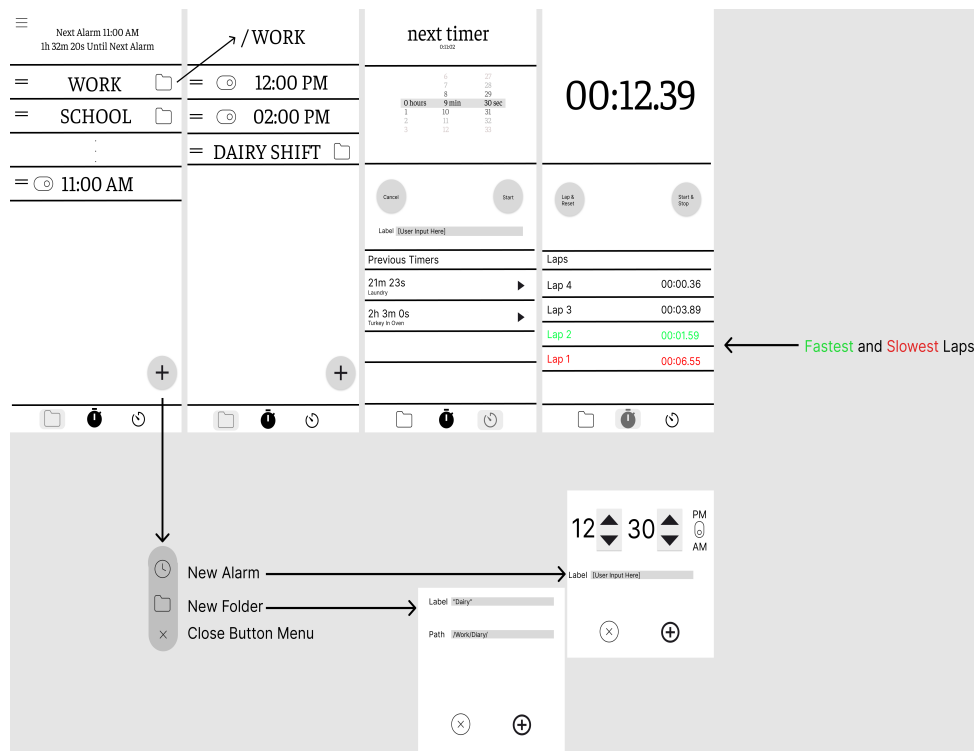


Figure 2: UI Mockup