

# Wake UP!!



Version 1.0 - 10/24/2024

Team Members:

Jooeun Jeon  
Daniel Macias  
Darpan Patel  
Rizwan Islam  
Yug Patel

Faculty Advisor:

Dr. Jung Soo Lim

# Table of Contents

1.	Introduction	2
1.1.	Background	2
1.2.	Design Principles	2
1.3.	Design Benefits	2
1.4.	Achievements	3
2.	Related Technologies	4
2.1.	Existing Solutions	4
2.2.	Reused Products	4
3.	System Architecture	5
3.1.	Overview	?
3.2.	Data Flow	?
3.3.	Implementation	?
4.	Conclusions	6
4.1.	Results	?
4.2.	Future	?
5.	References	7

# 1. Introduction:

## 1.1. Background:

WakeUP!! is an innovative application designed to ensure you start your day on time and remain awake once your alarm goes off. Traditional alarm clocks allow for easy snoozing. Our application introduces a unique and engaging approach that requires users to complete a mental challenge before they can disable the alarm. Our innovative method directly addresses the issue of people struggling to get out of bed in the morning and often falling back asleep, which disrupts their morning routine and negatively impacts their day.

Our team recognized that many individuals face significant challenges with the overuse of the snooze button. Hitting the snooze button repeatedly may seem harmless, but it can lead to increased fatigue, disrupt the body's natural circadian rhythm, and contribute to longer-term sleep-related health problems. These disruptions not only make it harder to wake up on time but also diminish the overall quality of sleep, leaving individuals feeling more tired and less productive throughout the day. WakeUP!! addresses these challenges by implementing a series of mental exercises that users must complete to turn off their alarms. These challenges are carefully designed to stimulate cognitive functions, thereby ensuring that users fully engage their minds upon waking. The application helps users transition more smoothly from sleep to awake, reducing the temptation to return to bed.

The mental challenges will range in difficulty and variety, ensuring that the user experience remains fresh and engaging. This not only makes the waking-up process more interactive and intentional. WakeUP!! provides an opportunity to establish a more disciplined and structured morning routine. Our application encourages users to wake up at a consistent time each day, which can reinforce the body's natural sleep-wake cycle and improve overall sleep quality. By reducing the likelihood of oversleeping and the negative impacts of snooze button abuse, WakeUP!! serves as a tool for enhancing both physical and mental well-being.

## 1.2. Design Principles:

- **User-Centered Design:** Providing a simple and clean interface to easily set alarms, customize sounds, and choose puzzles.
- **Cognitive Stimulation:** Ensuring that puzzles are challenging yet engaging to stimulate the brain upon waking.
- **Consistency:** Reinforcing healthy sleep habits by encouraging consistent wake-up times.
  - **Customizability:** Allowing users to tailor the app to their preferences, such as selecting puzzle difficulty or alarm sound.

### 1.3. Design Benefits:

- **Enhanced Wakefulness:** By solving puzzles, users fully engage their cognitive abilities, reducing the temptation to snooze or fall back asleep.
- **Health Benefits:** Encouraging consistent sleep schedules improves circadian rhythm, leading to better overall sleep quality.
- **Improved Productivity:** Starting the day with an active mind helps users transition smoothly to their daily activities.
- **Unique User Experience:** Combining mental challenges with a customizable alarm creates an engaging and practical application.

### 1.4. Achievements:

- Developed a seamless and responsive user interface using Flutter that adapts across devices.
- Implemented a puzzle-solving feature that integrates with the alarm system to ensure users are actively engaged upon waking.
- Successfully integrated notification functionalities using flutter\_local\_notifications.
- Enhanced app stability and performance with SQLite for alarm data storage.
- Designed the app to support cross-platform functionality, making it accessible on Android and iOS devices.

## 2. Related Technologies:

### 2.1. Existing Solutions:

We have looked into some existing frameworks for creating augmented reality visualization applications. These include Apple's ARKit, Google's ARCore, and Vuforia. While these solutions do allow us to reach our goal with this project, they come with certain restrictions and limitations.

Apple's ARKit is specific to Apple's iOS platform, and it is only available on version 11 of the operating system, with further restrictions on the different devices that can access this framework. So ARKit does not work for us if our intention is to reach a bigger audience.

Google's ARCore is in beta form, and it is very constrained in terms of performance. A very powerful Android device is necessary in order to run ARCore at reasonable performance. Furthermore, ARCore is only available on Android version 8.0 and upward, and again, much like Apple's ARKit, it is limited to only a few devices.

Vuforia is a reliable and proven multi platform AR framework. However, we decided against using it initially because of their approach to AR (as discussed in the next paragraph) and because there is a licensing cost.

The main approach of the AR frameworks mentioned above is to use Computer Vision techniques to recognize features in an image (such as the image from a smartphone's camera). When the device moves, the movement of the features on the screen can be detected. This allows the AR framework to render objects in the correct place on the screen, even as the device is moving.

This approach works well at detecting motion, but suffers when the registered features are not within sight (e.g. far away or obstructed by walls, trees or other objects). Also, while Computer Vision has come a long way at recognizing that a picture contains a building or mountain, these techniques have a hard time quickly identifying which building or mountain.

Since our AR framework is meant for representing distant specific objects, we need to use a different approach. Instead we are using GPS location and rotation of the device along with the coordinates of the object to be tracked to determine the proper place to draw objects on the screen. We did not find existing frameworks provided this functionality and thus were required to write our own.

## **2.2. Reused Products:**

2.2.1 Android Framework: The system leverages Android's native AlarmManager API for alarm scheduling and notifications. The system also utilizes Android's Material Design components for consistent UI/UX.

2.2.2 SQLite: The system uses SQLite for local data storage, such as alarm configurations and user preferences.

2.2.3 CAPTCHA Library: The system incorporates an existing CAPTCHA library to generate challenges for alarm dismissal. It uses the CAPTCHA library supports both text-based and image-based puzzles.

2.2.4 Android Accessibility Features: System utilizes Android's built-in accessibility tools, such as TalkBack and high-contrast mode. It complies with accessibility standards provided by Android to support users with disabilities.

## 3. System architecture:

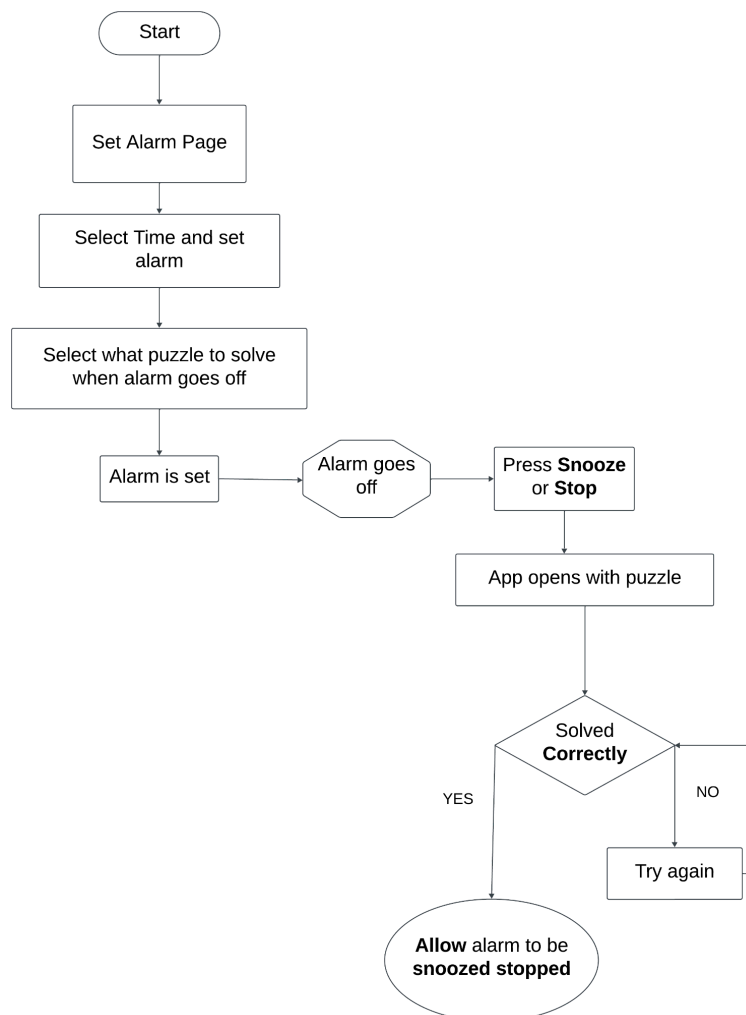
### 3.1. Overview:

The system architecture of WakeUP!! includes three main components:

1. **Frontend (UI/UX):** Built using Flutter, providing an intuitive and interactive user experience.
2. **Backend Logic:** Includes alarm management, puzzle generation, and notification handling using Dart and relevant libraries.
3. **Database Management:** Uses SQLite for storing user alarm configurations and history, ensuring data persistence and quick access.

### 3.2. Data Flow: (Add more to it)

Here is an overview of the Framework as a system:



The data flow of the alarm puzzle application begins when the user opens the app and is presented with the Set Alarm Page, where they select the desired alarm time and choose a puzzle type or difficulty level. Once the alarm is configured, it is programmed to activate at the specified time. When the alarm sounds, the user can either press Snooze, allowing for a brief delay before a puzzle challenge appears, or press Stop, which brings up the puzzle immediately. The user then attempts to solve the puzzle; if successful, the alarm will turn off, allowing the user to start their day. If the puzzle is not completed, the app prompts the user to try again, fostering engagement and encouraging cognitive activity from the outset. This structured flow ensures a seamless user experience that combines functionality with mental stimulation.

### **3.3. Implementation:**

The project was split into three sections to allow a well organize development: Designing the User Interface and User Experience, developing the core features, testing and deployment.

#### **3.4.1 Designing the User Interface and User Experience**

Our first step in implementing this application was to design an intuitive and engaging user interface (UI) and user experience (UX). Using Flutter's wide range of customizable widgets, we created a clean, user-friendly design that allowed users to easily set and manage their alarms, choose puzzle types, and customize alarm sounds. We also implemented fun visuals elements to keep users engaged, while maintaining simplicity so they could quickly access key features. Flutter's flexible and responsive layout system ensured that the design adapted seamlessly across devices, providing a consistent experience whether the app was used on a smartphone, tablet, or desktop. Streamlined navigation between alarm settings, history, and app settings.

## 4. Conclusions:

### 4.1. Results:

- WakeUP!! was successfully tested across multiple devices, ensuring smooth performance and compatibility.
- Feedback from beta testers highlighted the app's effectiveness in reducing oversleeping and improving morning routines.
- The integration of puzzles with alarms proved highly effective in engaging users and minimizing snooze abuse.

### 4.2. Future:

- **Enhanced Puzzle Variety:** Introduce additional puzzle types, such as memory games or logic puzzles, to cater to a wider audience.
- **Integration with Wearables:** Enable compatibility with smartwatches for alarm notifications and fitness tracking.
- **AI Personalization:** Use machine learning to recommend alarm settings and puzzles based on user habits.
- **Global Leaderboards:** Introduce a competitive aspect by allowing users to share and compare puzzle completion times with friends.



## 5. References:

Android Developer Website: <https://developer.android.com/reference/>

Vuforia Documentation: <https://library.vuforia.com/>

OpenGL es 2.0 standard:

[https://www.khronos.org/registry/OpenGL/specs/es/2.0/es\\_full\\_spec\\_2.0.pdf](https://www.khronos.org/registry/OpenGL/specs/es/2.0/es_full_spec_2.0.pdf)

OpenGL shading language 1.0 standard:

[https://www.khronos.org/registry/OpenGL/specs/es/2.0/GLSL\\_ES\\_Specification\\_1.00.pdf](https://www.khronos.org/registry/OpenGL/specs/es/2.0/GLSL_ES_Specification_1.00.pdf)

SQLite documentation: <https://www.sqlite.org/docs.html>