ADHD Task Manager

Non-Functional Requirements Specification

# 

# Introduction

The ADHD Task Manager is an application designed to address the challenges that people with attention deficit hyperactivity disorder (ADHD) face in managing their daily tasks and staying focused on their goals. Traditional task management and organization tools may not be effective for people with ADHD, as they require a high level of focus and discipline. The ADHD Task Manager offers a range of features, including a customizable task list, Pomodoro timer, habit tracker, rewards system, and insights and analytics.

These features are designed to be user-friendly and accessible, providing people with ADHD with an effective tool to manage their time and stay on track. Overall, the ADHD Task Manager is a valuable resource for people with ADHD who struggle with time management and organization, helping them to achieve their goals and improve their quality of life.

# System-Wide Functional Requirements

The ADHD Task Manager will be designed to address the challenges that people with ADHD face in managing their daily tasks and staying focused on their goals. To achieve this goal, the system will incorporate several key features, including a secure authentication process to ensure that only authorized users can access the system and their personal data.

In addition, the system will include task management functionality that allows users to create and manage their tasks easily. This functionality will enable users to add due dates, prioritize tasks, and categorize tasks by project or topic.

Customization will be a key feature of the ADHD Task Manager, allowing users to customize the interface to suit their preferences and needs. This functionality will include options to change the color scheme, font size, and layout of the application.

To help users stay focused and productive, the system will include a Pomodoro timer, which allows users to work in intervals of focused work and rest. The timer will be adjustable to accommodate different work preferences.

To encourage users to develop good habits and stay on track with their goals, the system will include a habit tracker. This functionality will allow users to set goals, track progress, and receive reminders to stay on track.

The ADHD Task Manager will also include a rewards system to motivate users to complete tasks and achieve their goals. This functionality will allow users to earn points or badges for completing tasks and provide incentives for continued use of the system.

Finally, to help users understand their productivity and track their progress, the system will include insights and analytics. This functionality will include graphs and charts that display data on completed tasks, time spent on tasks, and other relevant metrics.

By incorporating these system-wide functional requirements into the design and development of the ADHD Task Manager, the system will provide users with a comprehensive and effective tool for managing their time and staying on track with their goals.

# System Qualities

## Usability

The app will be designed to avoid cluttering by adhering to a minimalist design approach so that the screen is not distracting for users with ADHD. Moreover, the app will be designed with ease of navigation in mind.

The design pattern will be consistent to promote ease of using and learning about how the app works. In addition, verbal and/or text feedback, based on user preference, will be provided when the user interacts with the app. For instance, *List successfully created* or *Congratulations! Task AB Completed*.

To make the app for versatile and user friendly, multilanguage functionality will be added for localized support, which can include user manual, notifications, and instructions based on the users’ preference.

## Reliability

The reliability of the app will rely on factors such as successfully detecting errors like invalid user inputs and displaying appropriate error messages instead of crashing. Furthermore, the user will be given the option to enter correct inputs or quit.

Moreover, the reliability will be consolidated by thoroughly testing the app and fixing potential bugs. The app’s performance will also be monitored over time using analytics tools and error tracking software.

It is envisaged that the app will be able to work best with internet connectivity but will also provide limited services when there is no internet connection.

* App can work with/without internet.

## Performance

The performance of the app will be reflected on factors such as ability to scale and to accommodate the surge in the number of app users.

The apps performance will also be enhanced by

* Optimizing database queries. Such as not using SELECT\* but only columns needed.
* Aiming for a response time of <= 1 second.
* Using Android Profiler tool to measure response time.
* Using Android App Startup which is a library that helps optimize the startup process.
* Avoiding memory leaks to optimize shutdowns; use onDestroy() method to releases resources.
* the throughput will be identified by collecting performance data and analyzing performance metrics to understand the throughput.
* Reviewing the code as a group; peer reviewing.

## Supportability

Documentation

Support will be provided in various forms. The documentation component will include a multilanguage user manual that includes guidance to install the app. In addition, the documentation will provide explanations about the features included. A remote repository will be uploaded on GitHub with relevant documents which can support collaboration among developers.

To achieve flexible configurability, the system will be designed with modularity in mind for easier bug fixing. The app size will also be specified to help users make informed decisions.

Connectivity

It is envisaged that the app will function with and without internet connection. However, the latter will provide limited functions. Appropriate audio/visual messages will be displayed when offline so that users can know certain features/functions will not be accessible. This will allow users living in area with poor network connection to use the core features. The app will also connect with other preinstalled applications like Gmail to help keep the user organized.

Adaptability

The goal is to build the app with flexible configuration for greater adaptability which will include features such as:

* user changing font size, reminder tone, and frequency of reminders.
* users can change from auditory notifications to visual notifications and vice versa.
* using analytics to identify area where the app can be improved, such as features that are causing dissatisfaction to users.
  + - * Flurry analytics can be used to collect and store locally on device and once internet connection is available can send it to server such as Firebase analytics.

Localization

The goal is to provide a localized version of support from notifications to visual elements to improve user experience via closely resonating with a user’s local region and culture. This will be achieved by using a localization framework (Internationalization framework).

Moreover, appropriate graphics, images, and icons that aligns with users’ cultural background and preference will be used. Using a localization framework will also enable the app to be discovered in search engines using different languages.

App users can also be equipped with location awareness which can be used to notify them if their destination of interest is reached; For example, attending an appointment.

# System Interfaces

[Interface Requirements are part of the + in the FURPS+ classification of supporting requirements. Define the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and so forth, so that the software can be developed and verified against the interface requirements.]

## User Interfaces

[Describe the user interfaces that are to be implemented by the software. The intention of this section is to state requirements relating to the interface. Interface design may overlap the requirements gathering process.]

### Look & Feel

[Provide a description of the spirit of the interface. Your client may have given you particular demands such as style, colors to be used, and degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface.]

### Layout and Navigation Requirements

[Capture requirements on major screen areas and how they should be grouped together.]

### Consistency

[Consistency in the user interface enables users to predict what will happen. This section states requirements on the use of mechanisms to be employed in the user interface. This applies both within the system and with other systems and can be applied at different levels: navigation controls, screen areas sizes and shapes, placements for entering / presenting data, terminology.]

### User Personalization & Customization Requirements

[Requirements on content that should automatically displayed to users or available based on user attributes. Sometimes users allowed to customize the content displayed or to personalize displayed content.]

## Interfaces to External Systems or Devices

[Are there any external systems with which this system must interface? Are there any constraints on the nature of the interface between this system and any external system, such as the format of data passed between these systems, and any particular protocol used? Consider both provided and required interfaces.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so on.]

# Business Rules

[Business rules are statements that define or constrain some aspect of the business. Business rules are often represented as production rules when they are meant to be directly executed by an IT System: a production rule is an independent statement of programming logic that specifies the execution of one or more actions in the case that its conditions are satisfied. Production Rules define the operation semantic for the system in a technologic independent way. They constrain the behavior expressed in system use cases.

Organize this document on rule classes, a high level grouping of candidate or actual rules about one **business concept** with a specific kind of **logic processing**, example: Driver Risk Assessment Rules or Customer Validation Rules.]

## <Rule class name>

### <Rule name and ID>

[The description defines the rule. It can be made in natural language typically following a decision table or a pattern like: if [condition-list] then [action-list], example:

If there are at least 3 items of the same type in the customer shopping cart and each item’s value is greater than $30 then give to the customer a voucher whose value is 10% of the cheapest item.]

# System Constraints

Design Constraints

* + The app will be developed for Android operating system with the expected CPU [speed](https://insidetechworld.com/phones/top-android-phones-with-fastest-processors-in-2021/) of 1.8 GHz to 3 GHz or higher. The app is also expected to consume no more than \_\_\_ MB of flash memory and no more than \_\_\_ MB of system memory.

Implementation requirements

Android studio IDE will be used with Kotlin as the programming language and a distributed version control system will be used to allow simultaneous real-time collaboration and keep track of the project’s progress by keeping the relevant documents in a remote repository.

The documentations once completed and approved in the team meeting will be uploaded to the remote git repository. Group members will create individual local git branches and push commits to the remote branches to avoid breaking the main branch once coding begins. Google Maps SDK, a third-party component that can provide location-based reminders and notifications will be used.

* Google Maps SDK: An ADHD app could use the Google Maps SDK to provide location-based reminders and notifications. For example, the app could notify the user when they are approaching a specific location where they need to perform a task.

Interface requirements

With the modular approach in focus, the Model View Controller (MVC) design pattern will be followed. The view component will act as the presentation layer and it what the user will see on the screen. The model component will be the data layer, handling the business logic and data. The controller component will act as bridge between the other two components by processing user input, updating the model with new data and updating the view so that reflect the changes in the data.

Home screen must be easy to navigate and provide features such as a task list, timer, or a journal via a button.The app screen is also expected to follow the phones orientation, that is, landscape or portrait as per the phones orientation. The setting feature will allow users to customize such as font size, and notification setting.

The size and weight of the physical hardware on which the app will be hosted is expected to be identical to current Android smart phones in the market.

# System Compliance

## Licensing Requirements

The ADHD Task Manager will require a valid license for use. The license will be provided to the user upon the user registering an account and is valid until the user closes the account. The license will be enforced by the software, and users will be required to activate the software using their license key. The software will also include mechanisms to prevent unauthorized use of the system.

## Legal, Copyright, and Other Notices

The ADHD Task Manager will include legal disclaimers and copyright notices. The disclaimers will indicate that the software is provided "as-is" and that the developer makes no guarantees about the effectiveness or suitability of the software for any particular purpose. The copyright notices will indicate the owner of the software and the year of copyright. The software will also comply with any applicable trademark and logo compliance issues.

## Applicable Standards

The ADHD Task Manager will comply with industry standards for usability, interoperability, and internationalization. The software will be designed to work natively on Android devices. The software will also comply with any relevant legal and regulatory standards, including those related to data privacy and security. Additionally, the software will comply with any applicable industry standards for software development and testing, to ensure that the software is reliable and performs as expected.

# System Documentation

[Describes the requirements, for on-line user documentation, help systems, help about notices, and so on. Set expectations for the documentation and to identify who will be responsible for creating it.]