**1. Introduction**

**1.1 Purpose**

* Matzilon is a self-care utility app that aims to improve the mental stability, and provide a basic self-care to people with mental health issues.
* This document will describe the current notification system that Matzilon implements.

**1.2 Scope**

* The notifications are currently local in nature, as this is the first version of the feature and its most bare-bones version. Details on how to expand the system will be described in.

**1.3 Overview**

* The user is able to decide on what time to schedule his notification/s.
* In future implementation, maybe let the user pick what his notification should look like (either from a list of options or complete customization)

**2. System Overview**

**2.1 Background**

* In order to implement local notifications the package flutter\_local\_notifications was used.
* Currently the package supports both ios, android and web.
* We setup the system only for android. For further instructions on how to setup android and app look [here](https://pub.dev/packages/flutter_local_notifications).

**3. Requirements**

**3.1 Functional Requirements**

* משפט עידוד משתנה כל יום  
  מסך הגדרות -  
  בחירת השעה של ההודעה  
  אפשרות לבטל שליחת הודעות  
  לוקחים משפטים מ  
  inspirational quotes
* The user needs to allow us to use the USE\_EXACT\_ALARM and RECEIVE\_BOOT\_COMPLETED permissions.
* Those requirements weren’t implemented.

In order to fully implement this we need to move from local notification to push notifications.

Which in turn requires:

1. An external server to manage and send the notification.

2. A way to uniquely identify each user.

Without those 2 features the only possible implementation is a static massage.

**3.2 Non-Functional Requirements**

* The current local notifications do not require any additional, external requirements. But in order to fully implement the suggested system an authentication system needs to be implemented.

**4. Design Considerations**

**4.1 Assumptions**

* The user might want to schedule multiple notifications
* Editing the time is less important, as removing and adding a new one does the same thing.
* The user needs to know, and be able to cancel any and all upcoming notification

**4.2 Constraints**

* It’s not possible to send changing, truly dynamic, local notifications.
* We do not have an authentication system yet
* There is no external server to handle push notifications.

**4.3 Design Principles**

* The notification system is orthogonal to the app, and it’s implementation can be changed at any point with the minimal effort possible
* The notification screen was added to the sub menu where the other “extra features” are, as this is a more suiting place for it.

**5. System Architecture**

**5.1 High-Level Architecture Diagram**

* **Include diagrams** showing system components and interactions.

**5.2 Components Description**

The menu now has a notification page.

The notification page has a time-picking area, in where the user picks the time for the notification.

It also has a Notification List which is a list of Notification List Items, detailing the scheduled notifications and allowing the user to cancel them.

The notification service itself is implemented in notification\_service and is called NotificationHelper

**6. Detailed Design**

**6.1 Module Overview**

**6.1.1 notification\_service**

* **Purpose**: implements the notification system, using the package flutter\_local\_notifications.
* **Functionality**: let’s the user add and remove a notification(currently only on android)
* **Interfaces**: NotificationHelper.init() initializes the notification service  
  NotificationHelper.scheduleNotification( TimeOfDay time, String id) schedules a notification with the identifier id at the time time

NotificationHelper.cancelNotification(String id) cancels the notification with the given id

NotificationHelper.cancelNotifications() resets the system and cancels all notifications.

**6.1.1 notification\_list\_item**

* **Purpose**: create each notification widget to display to the user.
* **Functionality**: shows a notification and allow the user to delete it.

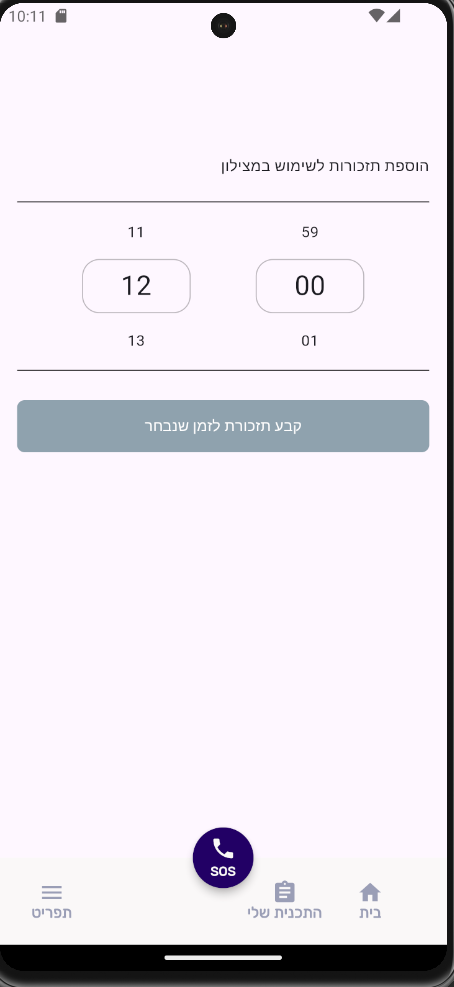
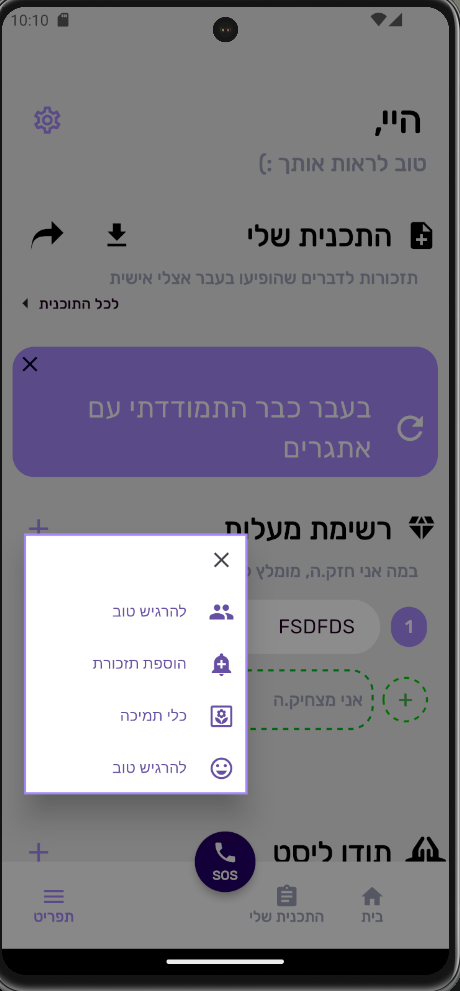
**6.1.1 notification\_list**

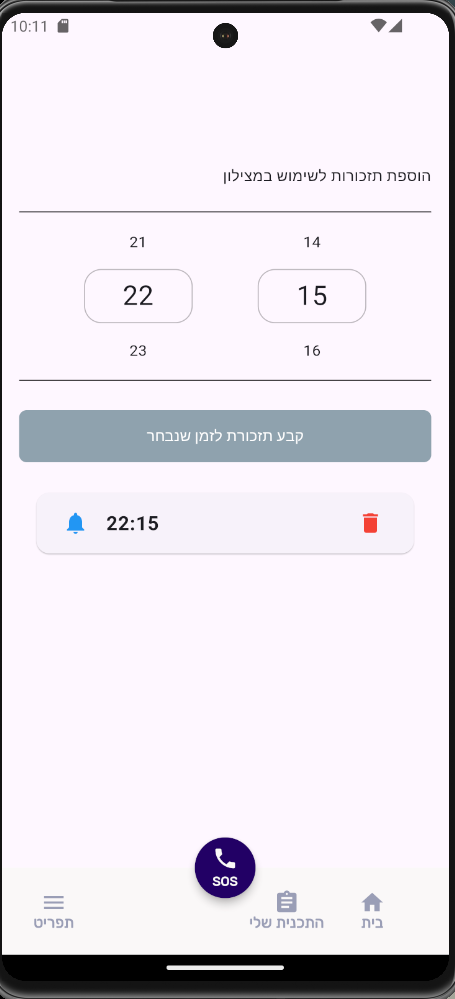
* **Purpose**: show to the user the list of all current notifications.
* **Functionality**: displays the list of notifications

**6.1.1 notification\_page**

* **Purpose**: creates the interface for the user to manage his notifications
* **Functionality**: lets the user pick at which time to schedule a notification and manage his notifications.

**8. User Interface Design**

Here is the 0.1 UI version  




**9. External Interfaces**

**9.2 Software Interfaces**

* For flutter\_local\_notifications:
* FlutterLocalNotificationsPlugin() creates the api that lets us interact with the package
* FlutterLocalNotificationsPlugin.Initialize() lets us setup and start the notification system for all systems
* InitializationSettings() lets us customize the notification’s visual component
* FlutterLocalNotificationsPlugin.zonedScheduled() schedules a repeating notification based on the parameters given to it

For more details look [here](https://pub.dev/packages/flutter_local_notifications).

**10. Security Considerations**

**10.1 Authentication**

* Like described before, for the full implementation of the desired system an authentication system needs to be implemented.