POLITECNICO DI MILANO

MSc Automation and Control Engineering

Software Engineering

Testing Document

For the Application of Medicine Reminder

Prof. Matteo Giovanni Rossi

Avinash Umesh Sarma 951410
Dineshkumar Ramasamy 964069
Satyabrata Dash 961528

ACADEMIC YEAR 2021/2022

Contents

1. Introduction

From studying the requirements for the medicine reminder, the Feasibility Study document provides evident justifications for developing the same. The development of the mobile applications follows different phases of documentation, starting from the Feasibility Study Document which explains the feasibility of the concept. Further, the next phase involves the study of requirements to develop the concept that makes the Requirement Analysis and Specifications Document (RASD) while the study of software architecture to follow is enclosed in the Design Document (DD). The other phases include the Implementation Document, Testing Document, and User Manual, where the information regarding the documents would be mentioned correspondingly in the later phases of documentation.

Here, this illustrates the Testing Document (TD) of the project of the medicine reminder abbreviated and named "MedRem". The document particularly focuses on the testing aspect of the application with the framework adopted. Thus, the document starts with a section on the testing strategy and test plan, which also gives an overview of the testing of different features. In the later section a use-case testing is performed on certain features. Furthermore, this document encloses the UI/UX testing of different screens for the users to follow, to use the mobile application.

1.1 Scope

MedRem is an easy-to-use application that helps people to keep track of their medicines by providing reminders. It has a simplistic design and vintage approach to the features. This application consists of adding, showing, modifying, and deleting the medicine information. Other than that, two kinds of reminders can be added to the medicines, namely a medicine reminder and a stock reminder. The user can set and modify them according to their intake of the medicines. Restock is yet another feature that is added to the application, allowing the user to update the current stock value.

1.2 Assumptions and Constraints

The product's assumptions are taken into consideration while testing:

a) The hardware (mobile phone) and the software (OS) are compatible with the application.
b) The application is not suitable for varying screen sizes.
c) The application is not functional on a power-saving mode.
d) The testing of various functionalities of the application is compiled considering one medicine.

2. Testing Strategy

The application undergoes black box testing. The black box is a powerful technique to check the application under test from the user's perspective. Black box testing is used to test the system against external factors responsible for software failures. This testing approach focuses on the input that goes into the software, and the output that is produced. The testing does not cover the inside details such as code, server logic, and development method.

Black box testing is based on the requirements and checks the system to validate against predefined requirements.

The purpose of the black box testing procedure is to derive test cases from the specifications. Within the black box testing, the application is tested by performing the functional black box testing which concerns only the functional requirements of the system and covers how well the system executes its functions.

The actions and inputs for different modules follow the boundary value analysis and equivalence partitioning. While performing the test scenarios, a positive test case, where the user can perform appropriate actions when using valid data, and a negative test case, which is performed to try to "break" the application by performing invalid or unacceptable actions or by using invalid data are also considered.

3. Test Plan

Module testing: In this section, the application is being tested for individual modules to determine all the technical aspects and objectives of the requirements. The modules under testing are:

a) Medicine reminder
b) Alarms and Notifications
c) Stock reminder

Integration Testing: In this section, the application is being tested for individual features, and then with an increase in complexity, the testing of the integration of different modules is also carried out to extract the technical aspects and objectives of the requirements. Given below are the operations under test.

a) Add medicine
b) Modify medicine
c) Delete medicine
d) Show medicine

Use Case Testing: This section will be representing the real-time user-created pathway using different functionalities provided by the application. Use Case Testing is a software testing technique that helps to identify test cases that cover the entire system on a transaction-by-transaction basis from start to end. Test cases are the interactions between users and software applications. Use case testing helps to identify gaps in the software application that might not be found by testing individual software components. It includes use cases such as:

a) Add medicine details
b) Show medicine details
c) Delete medicine details
d) Restock medicine details

Important: The test plan was carried out on the important test cases with respect to the application.

3.1. Module Testing

3.1.1. Medicine Reminder Module

While performing the testing, states like 'launch application' or the 'screen changes' are not considered here. Only the module functionality is tested.

The visibility of the medicine reminders time is dependent on the frequency and state of the switch.

Test Scenario ID	Med-1	Test Case ID	Med-1A
Test Case Description	Medicine reminder-positive test case	Test Priority	High
Pre-Requisite	The medicine details must be filled	Post-Requisite	NA

Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Set the medicine reminder.	The user toggles ON the medicine reminder switch.	The switch changes its state and lists out the time pickers	The switch changes its state and lists out the time pickers	Pass	The switch is in perfect condition
2	Set the medicine reminder.	The user toggles the medicine reminder switch ON and sets	Reminder added successfully	Reminder added successfully	Pass	Data entered successfully

		the time at 0900 and 1600				

Test Scenario ID	Med-1	Test Case ID	Med-1B
Test Case Description	Medicine reminder-negative test case	Test Priority	High
Pre-Requisite	The medicine details must be filled	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Set the medicine reminder.	The user toggles the medicine reminder button ON, and choosing same time for different intakes	Invalid time. Please give a different time.	Reminder added successfully.	Fail	Data entered successfully
2	Leave the 'choose time' as empty as per the frequencies.	The user toggles the medicine reminder button ON only.	Please set the intake time. .	Please set the intake time.	Pass	Data compared successfully

3.1.2. Alarms and notifications module

Test Scenario ID	Alarm-1	Test Case ID	Alarm-1A
Test Case Description	Alarm and notifications-positive test case	Test Priority	High
Pre-Requisite	NA	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	The user adds a medicine with the medicine reminder.	The user toggles ON the medicine reminder switch and sets a unique time.	The alarm goes off and the notification arrives.	The alarm doesn't go off, but the notification arrives.	Fail	The alarm doesn't work
2	The user adds a medicine with the stock reminder.	The user toggles ON the stock reminder switch and sets a value for	The alarm goes off and the notification arrives.	The alarm doesn't go off, but the	Fail	The alarm doesn't work

		No. of pills and Minimum quantity.		notification arrives.		

Test Scenario ID		Alarm-1		Test Case ID	Alarm-1B	
Test Case Description		Alarm and notifications-negative test case		Test Priority	High	
Pre-Requisite		NA		Post-Requisite	NA	
Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	The user adds a medicine with the medicine reminder.	The user toggles ON the medicine reminder switch and sets the same time.	Invalid Time. Thus, no alarm and notification	Time saved The alarm fails but the notification works	Fail	The alarm doesn't work
2	The user adds a medicine with the stock reminder.	The user toggles ON the stock reminder switch and sets a negative value for No. of pills and Minimum quantity.	Invalid data entry. Thus, no alarm, and the notification	Data added successfully. The alarm doesn't go off, but the notification arrives.	Fail	The alarm doesn't work

3.1.3. Stock Reminder module

The visibility of the stock reminders fields like no. of pills and minimum quantity is dependent on the dosage and state of the switch.

Test Scenario ID		Stock-1		Test Case ID	Stock-1A	
Test Case Description		Stock reminder-positive test case		Test Priority	High	
Pre-Requisite		The medicine details must be filled		Post-Requisite	NA	
Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Set the stock reminder.	The user toggles ON the stock reminder switch.	The switch changes its state and lists	The switch changes its state and lists	Pass	The switch is in perfect condition

			out the fields like No. of Pills and Minimum quantity	out the fields like No. of Pills and Minimum quantity		
2	Set the stock reminder.	The user toggles the stock reminder button ON, and No. of Pills is set to 20. The minimum quantity is set to 5	Reminder added successfully	Reminder added successfully	Pass	Data entered successfully

Test Scenario ID	Stock-1	Test Case ID	Stock-1B
Test Case Description	Stock reminder-negative test case	Test Priority	High
Pre-Requisite	The medicine details must be filled	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Set a stock reminder.	The user toggles the stock reminder button ON, and No. of Pills is set to 15. The minimum quantity is set to 30	No. of pills should be greater than the minimum quantity	Please enter a value for the stock more than the minimum quantity	Pass	Data compared successfully
2	The fields in the stock reminders are empty.	The user toggles the stock reminder button ON. No. of pills is left blank, and the minimum quantity is also left blank.	Please enter the required fields.	Please enter the required fields.	Pass	Data compared successfully
3	Set a stock reminder.	The user toggles the stock reminder button ON, and No. of Pills is set to -5.	Please enter a value for the stock more than the minimum quantity	Reminder added successfully	Fail	Data compared successfully

		The minimum quantity is set to -10				

3.2. Integrated Module test cases:

3.2.1. Add Medicine operation

While performing the testing, states like 'launch application' or the 'screen changes' are not considered here. Only the module functionalities are tested.

The testing completely focuses on adding the medicine to the application. So, there is no inclusion of any kind of reminder.

Screen 2 and 3 is associated with this operation along with the usage of modules such as reminders and database in context to the UML diagram of the design document.

Test Scenario ID	Add-1	Test Case ID	Add-1A
Test Case Description	Add Medicine-positive test case	Test Priority	High
Pre-Requisite	NA	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Enter a valid name, frequency, and dosage of the medicine	Name: Pantoprazole Frequency: 2 Dosage:2	Data added successfully	Data added successfully	Pass	Data entered successfully

Test Scenario ID	Add-1	Test Case ID	Add-1B
Test Case Description	Add Medicine-negative test case	Test Priority	High
Pre-Requisite	NA	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Enter an invalid name. The frequency	Name: 48 () &5 6+ Frequency: 2 Dosage:2	Invalid name. Please re-	Data added successfully	Fail	Invalid Data entered successfully

	and dosage are valid.		enter a valid name			
2	Enter a negative dosage value. The frequency and name of the medicine are valid.	Name: Pantoprazole Frequency: 2 Dosage: -10	The dosage value is invalid. Please re-enter the dosage value.	Data added successfully	Fail	Invalid Data entered successfully
3	Leave the medicine fields empty.	Name: Dosage: Frequency:	Please enter the required fields.	Please enter the required fields.	Pass	Data compared successfully
4	Enter the same name, frequency, and dosage of a previously added medicine.	Name: Pantoprazole Frequency: 2 Dosage:2	Medicine already exists	Medicine already exists	Pass	Data compared successfully

3.2.2. Show Medicine operation

The testing completely focuses on showing the medicine in the application. Like the previous functionality, this feature didn't include any positive or negative test cases because there aren't any valid or invalid inputs. The action could be performed in a single step.

When the show medicine feature is tested, the user could only provide a valid input which would be to view the details of the medicine. The only exception comes in this feature when the user didn't add any medicine to the application and tries to view any medicine. At that point, the user receives a friendly notification to add medicines.

Screen 4 is associated with this feature along with modules namely listpicker, database, and display in context to the UML diagram of the design document.

Test Scenario ID	Show-1	Test Case ID	Show-1A
Test Case Description	Show Medicine	Test Priority	High
Pre-Requisite	A medicine should already be added.	Post-Requisite	NA

Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Results	Test Comments

1	Show selected Medicine	The user clicks 'show medicine'	Medicine names are displayed in form of a list	Medicine names are displayed in form of a list	Pass	Data viewed successfully

3.2.3. Delete Medicine operation

The testing completely focuses on deleting the medicine in the application. Like Show Medicine functionality, this feature also didn't include any positive or negative test cases because there aren't any valid or invalid inputs. The action could be performed in a single step.

When the delete medicine feature is tested, the user could only provide a valid input which would be to delete the details of the medicine. The only exception comes in this feature when the user didn't add any medicine to the application and tries to access this feature. At that point, the user receives a friendly notification to add medicines.

Screen 4 is also associated with deleting medicine as it is a part of the show medicine screen along with the modules namely listpicker, database, and display in context to the UML diagram of the design document.

Test Scenario ID	Del-1	Test Case ID	Del-1A
Test Case Description	Delete Medicine	Test Priority	High
Pre-Requisite	A medicine should already be added.	Post-Requisite	NA

Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Results	Test Comments
1	Delete selected Medicine	The user clicks 'delete'	Data deleted successfully	Data deleted successfully	Pass	Data deleted successfully

3.2.4. Modify medicine operation

The testing completely focuses on the modification of the medicine in the application. It is performed only when medicine is added to the application.

Screen 5 is associated with the operation with the inclusion of modules like reminders, database, and display in context to the UML diagram of the design document.

Test Scenario ID	Mod-1	Test Case ID	Mod-1A
Test Case Description	Modify Medicine-positive test case	Test Priority	High
Pre-Requisite	A medicine should already be added.	Post-Requisite	NA

Test Execution Steps Results

S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Modify the fields like name, frequency, and dosage of the medicine according to the inputs.	Previous input: Name: Azibact Frequency: 1 Dosage:1 New input: Name: Azithromycin Frequency: 3 Dosage:3	Medicine modified successfully	Medicine modified successfully	Pass	Data modified successfully
2	Set medicine reminder for a previously added medicine.	The user toggles the medicine reminder switch ON and sets the timer value.	Medicine modified successfully	Medicine modified successfully	Pass	Data modified successfully
3	Set the stock reminder for a previously added medicine.	The user toggles the stock reminder switch ON and sets the value for no. of pills and minimum quantity.	Medicine modified successfully	Medicine modified successfully	Pass	Data modified successfully
4	Modify the no. of pills through restock option for a previously added medicine.	The user clicks restock button and enter No. of pills as 10.	Data modified successfully	Data modified successfully	Pass	Restock performed successfully

Test Scenario ID	Mod-1	Test Case ID	Mod-1B
Test Case Description	Modify Medicine-negative test case	Test Priority	High

Pre-Requisite		A medicine should already be added.		Post-Requisite	NA	
Test Execution Steps Results						
S.No	Action	Inputs	Expected Output	Actual Output	Test Result	Test Comments
1	Modify the field name and dosage of a medicine.	Name: A456$%^-= Frequency: 3 Dosage: -1	Invalid Name. Please re-modify it. The dosage value is invalid. Please re-enter a value.	Name: A456$%^-= Frequency: 3 Dosage: -1	Fail	Invalid Data modified successfully
2	Modify the field of frequency to empty and valid name and dosage.	Name: Azithromycin Frequency: Dosage: 2	Please choose frequency	Please choose frequency	Pass	Data modified successfully
3	Set the medicine reminder for a previously added medicine.	The user toggles the medicine reminder button ON and enters First intake time: 08:00 Second intake time: 08:00 Third intake time: 08:00	Time values are coinciding. Please change the time values.	Data modified successfully.	Fail	Medicine Reminder data modified successfully
4	Set the medicine reminder without choosing a time.	The user toggles the medicine reminder button ON only.	Please set the time	Please set the time	Pass	Data compared successfully
5	Set the stock reminder but with negative values.	The user toggles the stock reminder button ON, and No. of Pills is set to -5.	Invalid values. Please re-enter.	Reminder added successfully	Fail	Data compared successfully

		The minimum quantity is set to -10				
6	The stock reminder fields are left empty.	The user toggles the stock reminder button ON. No. of pills is left blank, and the minimum quantity is set to blank	Please enter the required fields.	Please enter the required fields.	Pass	Data compared successfully
7	Update the current stock value through restock.	The user enters No. of pills as -10.	No. of pills cannot be a negative number	Data modified successfully.	Fail	Restock performed successfully

3.3. Use Case Testing

3.3.1. Add Medicine with reminders

Use Case Name	Add Medicine with the reminders	
Use case Description	A user wants to add the medicine details with setting the respective reminders to the application.	
Actors	Users	
Pre-Condition	Application must be opened by the user.	
Post -Condition	After a successful addition, a notification is displayed.	
Main Scenarios	Serial No	Steps
A: Actors/Users S: System	1	User clicks 'Add Medicine'
	2	System directs to next screen
	3	User enters the name, dosage, and frequency of the medicine and clicks 'Set Reminder'
	4	System directs to next screen.
	5	User toggles ON the medicine reminder switch.
	6	System displays the fields to choose the time.
	7	User chooses the time
	8	System displays the time set
	9	User toggles ON the stock reminder switch
	10	System displays the fields
	11	User enters the no. of pills and minimum quantity and clicks 'Save'
	12	System adds the medicine to the database.

Extensions	3a	Empty field ‘name’, ‘dosage’, and ‘frequency’ S: shows a notification to fill them.
	11a	Empty fields ‘time’, ‘no. of pills’, and ‘minimum quantity’ S: shows a notification to fill them
	11b	No. of pills is less than the minimum quantity S: shows a notification to modify them

3.3.2. Show Medicine details

Use Case Name	Show Medicine	
Use case Description	A user wants to view the medicine details in the application.	
Actors	Users	
Pre-Condition	Application must be opened, and the user must be on the home screen.	
Post -Condition	Users views the detailed information on the added medicine.	
Main Scenarios	Serial No	Steps
A: Actors/Users S: System	1	User clicks ‘Show Medicine’.
	2	System displays all the medicine names in the form of a list.
	3	User selects a particular medicine.
	4	System displays the detailed information on the medicine.
Extensions	1a	If the medicine is not added. S: shows a notification to add medicine.

3.3.3. Delete Medicine details

Use Case Name	Delete Medicine	
Use case Description	A user wants to delete the medicine details in the application.	
Actors	Users	
Pre-Condition	Application must be opened, and the user must be on the home screen.	
Post -Condition	After a successful deletion, a notification is displayed.	
Main Scenarios	Serial No	Steps
A: Actors/Users S: System	1	User clicks ‘Show Medicine’.
	2	System displays all the medicine names in the form of a list.
	3	User selects a particular medicine.
	4	System shows detailed information on the medicine added.
	5	User clicks ‘Delete’

	6	System deletes the medicine from the database and gives a notification.
Extensions	1a	If the medicine is not added. S: shows a notification to add medicine.

3.3.4. Restock Medicine details

Use Case Name	Restock Medicine	
Use case Description	A user wants to restock the medicine in the application.	
Actors	Users	
Pre-Condition	Application must be opened, and the user must be on the home screen. The stock reminder must be added to the medicine prior.	
Post -Condition	After a successful modification, a notification is displayed.	
Main Scenarios	Serial No	Steps
A: Actors/Users S: System	1	User clicks 'Show Medicine'.
	2	System displays all the medicine names in the form of a list.
	3	User selects a particular medicine.
	4	System shows detailed information on the medicine added.
	5	User clicks 'Modify'
	6	System takes to the next screen.
	7	User clicks 'Restock'
	8	System displays a notification for entering the value
	9	User enters the value and presses OK
	10	System saves the modification in database
Extensions	1a	If the medicine is not added. S: shows a notification to add medicine.
	9a	If the field is empty S: shows a notification to fill the field.
	9b	If the input is invalid S: shows a notification to change the input.

4. UI/UX Testing

4.1. Home Screen

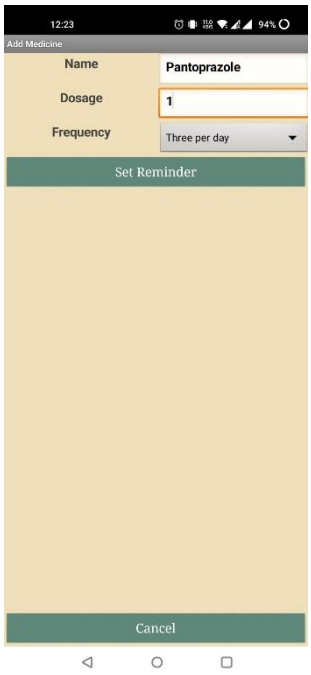
Spelling: On the home screen, there are no typographical errors.

Cosmetic Inconsistencies: On the home screen, there are no cosmetic inconsistencies.

Spacing and Alignment: For the home screen, the image is perfectly centered and spacing between the image, application name, and buttons are also uniform.

Interaction: Both the 'add medicine' and 'show medicine' are clickable buttons. Show medicine shows a notification to 'Please add medicine' when there are no added medicines.

4.2. Add Medicine screen



Required fields: Fields like name, dosage, and frequency are mandatory fields. Identification of the mandatory fields is not done either using a red asterisk or text, yet they give a friendly warning to fill the empty fields.

Data types: The 'Name' field can take a combination of upper-case and lower-case letters, numbers, and special characters. 'Dosage' field is restricted to numbers only. 'Frequency' is an enumeration that has the choice to provide up to three times a day.

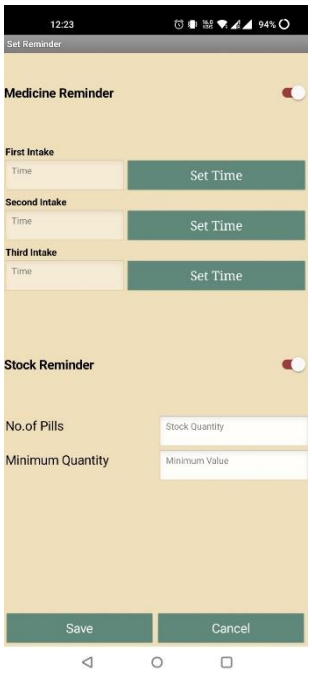
Spelling: On the add medicine screen, there are no typographical errors.

Cosmetic Inconsistencies: On the add medicine screen, there are no cosmetic inconsistencies.

Spacing and Alignment: For this screen, the text fields are perfect and the spacing between the fields and buttons is also uniform.

Interaction: The 'Set Reminder' and 'Cancel' are clickable buttons and perform their respective tasks assigned.

4.3. Set Reminder screen



Required fields: The 'set time' for all the intakes in the medicine reminder and the 'No. of Pills and 'Minimum quantity' in the stock reminder are mandatory fields on the screen. Identification of the mandatory fields is not done either using a red asterisk or text, yet they give a friendly warning to fill the empty fields.

Data types: The 'No. of pills' and 'Minimum quantity' fields in the stock reminder take only numerical values. The time display of the medicine reminder is in 24-hour format.

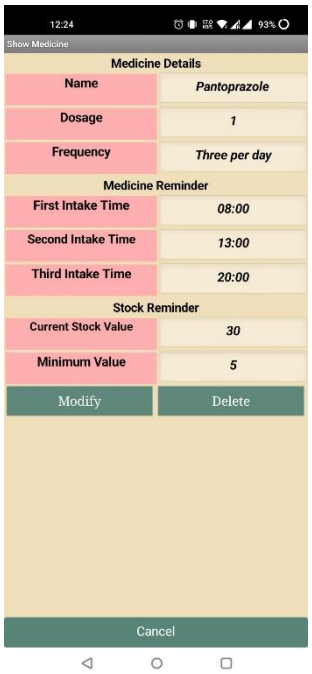
Spelling: On the set reminder screen, there are no typographical errors.

Cosmetic Inconsistencies: On the set reminder screen, there are no cosmetic inconsistencies.

Spacing and Alignment: For this screen, the switch and the set time field are perfect and the spacing between the fields and buttons is also uniform.

Interaction: Both the medicine reminder and stock reminder switches toggle ON and OFF. When toggled ON, it allows the user to enter the values. Even after multiple toggling of both the switches, their respective values remain the same. The 'Save' and 'Cancel' are clickable buttons and perform their tasks without fail.

4.4. Show Medicine screen



Data types: All the data types are displayed correctly.

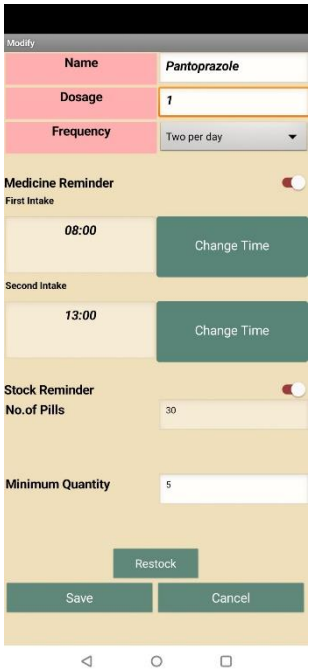
Spelling: On the Show medicine screen, there are no typographical errors.

Cosmetic Inconsistencies: On the show medicine screen, there are no cosmetic inconsistencies.

Spacing and Alignment: For this screen, the text fields are perfect and the spacing between the fields and buttons is also uniform.

Interaction: The 'Modify', 'Delete', and 'Cancel' are clickable buttons and perform their tasks respectively.

4.5. Modify Medicine Screen



Required fields: Fields like name, dosage, frequency, No. of Pills and Minimum quantity are mandatory fields. Identification of the mandatory fields is not done by a red asterisk (*).

Data types: The 'Name' field can take a combination of upper-case and lower-case letters, numbers, and special characters. 'Dosage', No. of pills, and the Minimum quantity fields are restricted to numbers only. 'Change time' also follows the 24-Hour format.

Spelling: On the Modify medicine screen, there are no typographical errors.

Cosmetic Inconsistencies: On the Modify medicine screen, there are no cosmetic inconsistencies.

Spacing and Alignment: For this screen, the text fields are perfect and the spacing between the fields and buttons is also uniform.

Interaction: Both the medicine reminder and stock reminder switches toggle ON and OFF. When toggled ON, it allows the user to modify the previous values. Even after multiple toggling of both the switches, their respective values remain the same.

The 'Restock' is a clickable button. Without adding a stock reminder, it won't work. Once the reminder is added, clicking it, the system asks for user input value for the new stock to be added to the old stock.

The 'Save' and 'Cancel' are clickable buttons and perform their respective tasks without fail.

5. References

- https://www.javatpoint.com/testing-documentation
- https://www.guru99.com/testing-documentation.html
- https://www.softwaretestinghelp.com/test-case-template-examples/
- https://netmind.net/en/positive-vs-negative-test-cases/#:~:text=Positive%20test%20cases%20ensure%20that,or%20by%20using%20invalid%20data.
- https://www.guru99.com/use-case-testing.html
- https://www.softwaretestinghelp.com/use-case-testing/
- http://tryqa.com/what-is-use-case-testing-in-software-testing/
- https://www.resene.co.nz/homeown/use_colr/colours-for-living.htm