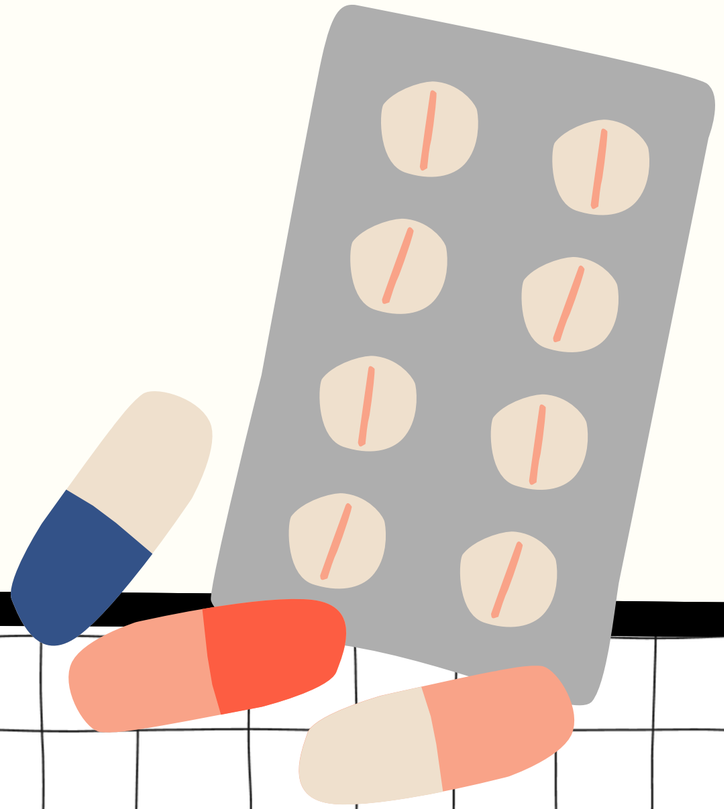
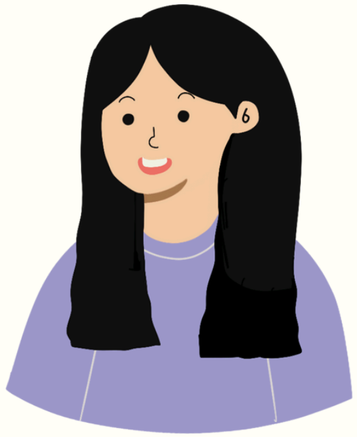
Project Proposal

MEDICATION SCHEDULER

Prepared by: Group 9



The team



NUR ARISHA

ALYA QISTINA

TANG JASMINE

General Idea

Our proposed system is the medication scheduler. The general idea of this system is to develop a system which can be used by individuals who have to take medication on a schedule with the accurate dosage, especially for those who have to take different medications with different dosage at a time. This is an upgraded version of the traditional system that uses labeled containers to alert patients on medicine intake. Instead, this system can be integrated into their device and can be accessed at any time.

System objectives and/or purpose

What the user can achieve/do in the system?

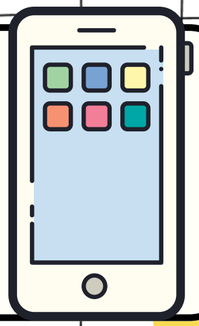
Medication scheduling:
record prescribed medication details

Keep track:
dosage, timing, routine and progress

Reminder:
Alert user the time to take medicine by send notification

Secondary assurance:
supervision from guardian or personal doctor

Portable :
offers the convenience of having all medication information in one place (system accessible through any electronic device: smart watch/phone)

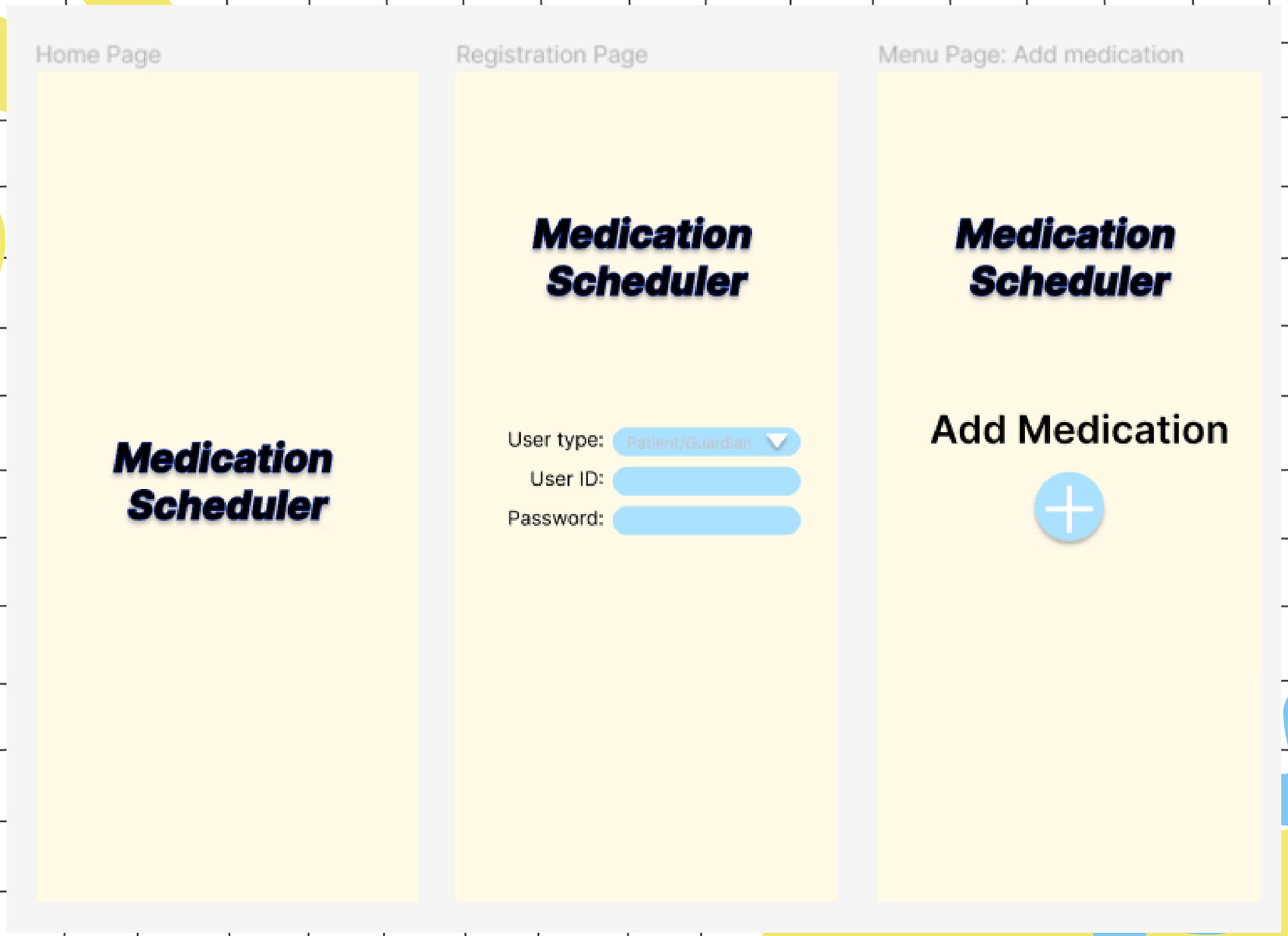


How to use the system

Medication Scheduler STEP BY STEP USE:

1) Open app
2) Register you and your private doctor account
3) Click on "ADD MEDICATION" button, this will bring you to the next page
4) Medication name will be requested. Type the name of your medication, more precise the better. Click "NEXT"
5) A page to choose the type of your medication will pop up. It is a multiple choice.
-CAPSULE, TABLET, LIQUID, CREAM, DROPS, INJECTION, LOTION, GEL, OINTMENT, SPRAY, PATCH
Choose the one most identical to your medication. Click "NEXT"
6) You will be brought to a page to insert your medication strength. Insert it as well as the unit (mg, mcg, g, ml, %). Click 'NEXT'
7) You will be asked the frequency of the use of medication. You also can add the time. Once this is done, you can ensure you will be notified to take your medication. Click "NEXT"
8) You can opt to insert any additional details or just skip this. Click "DONE"
9) A calendar will pop up, showing when you must take the medication.

•



How to use the system

Menu Page: Medication name

Medication Scheduler

Medication Name

Add Medication Name

Next

Menu Page: Medication type

Medication Scheduler

Medication Type

Capsule

Tablet

Liquid

Cream

Drops

Injection

Lotion

Gel

Ointment

Next

Menu Page: Medication strength

Medication Scheduler

Medication Strength

Add Strength

Unit

Add Unit (mg, mcg, g, ml, %)

Next

How to use the system

Menu Page: Schedule Medication

Medication Scheduler

Frequency

Daily/Weekly etc.

Time

Add time

Next

Menu Page: Review Schedule M...

Medication Scheduler

Schedule

Frequency

Time

Next

Menu Page: Medication Details

Medication Scheduler

Shape

Add shape

Color

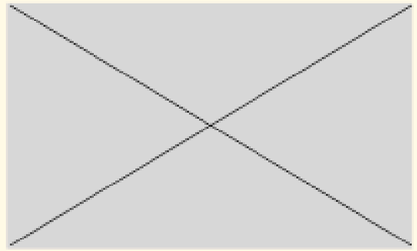
Add color

Done

Menu Page: Calendar

Medication Scheduler

Calendar



Add Medication

How to use the system

Example of Outputs

Table categorizing the medicine name, forms, dosage, frequency of administration and time to be taken.

Report of patients' medical history.

Copy of medication intake to guardian/in-house doctor.

Alert when patients skipped their medication.

Array of Object

Parent class	Child class	Member
User	Owner, Guardian	UserID, Password
Reminder	Medication, Dosage	MedID, noDosage
Time	Frequency, Routine	Date, time

Encapsulation

In general, encapsulation may be defined as the process of wrapping up data and information in a single unit. In C++ Object Oriented Programming is combining and binding data and function that manipulate it together.

How to do encapsulation:

1. Make all the data contained as a private members
2. Create an accessor and mutators for each data member that were created in private

Example:

Using our project as an example to explain the details of encapsulation, in a medication scheduler app, there exist two different users, the patient, and the private doctor that specifically only that patient.

Patient: They can use the app and access only their account. They also may schedule their medication intakes, make appointments, input medication's type, dosage, and also set reminders to take their prescription.

Doctor: The app allows doctors to view their patients' medication dosage and make recommendations or adjustments as needed. They can also prescribe new medication or update current ones.

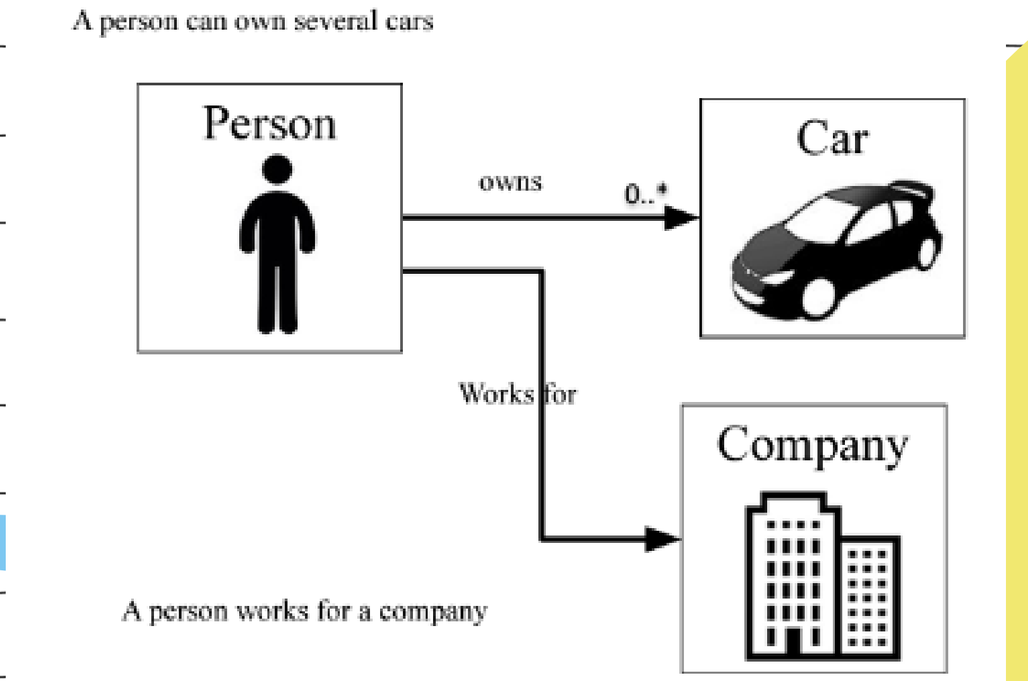
However, due to encapsulation, a patient can only access and see their own medication regimen. They are unable to view other patient's account or make changes to the app functionality. Similarly, doctors too can only see and manage their patients prescription schedule. They cannot access other patient's data.

In this scenario, encapsulation plays a crucial role in protecting the data and system. It helps maintain data privacy and security. It restricts access to sensitive data.

Association

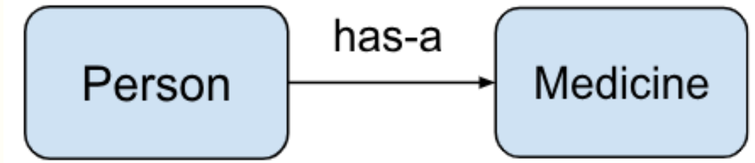
The general idea of association is objects can be related to each other in a class. The relationship between objects can be:

- One-to-many
- One-to-one
- Many-to-one
- Many-to-many



Aggregation

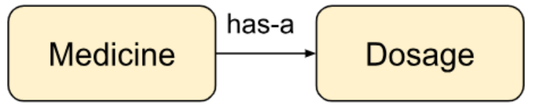
A weaker relationship than composition.
Responsible for containing the object, but not for its creation nor destruction.
Can exist independently.



(object Person can exist without Medicine)

Composition

A stronger relationship than regular association.
One class is the owner of another class (in a relationship) and responsible for its creation and destruction.
If the containing object is destroyed, the contained object will also be destroyed
Cannot exist independently.

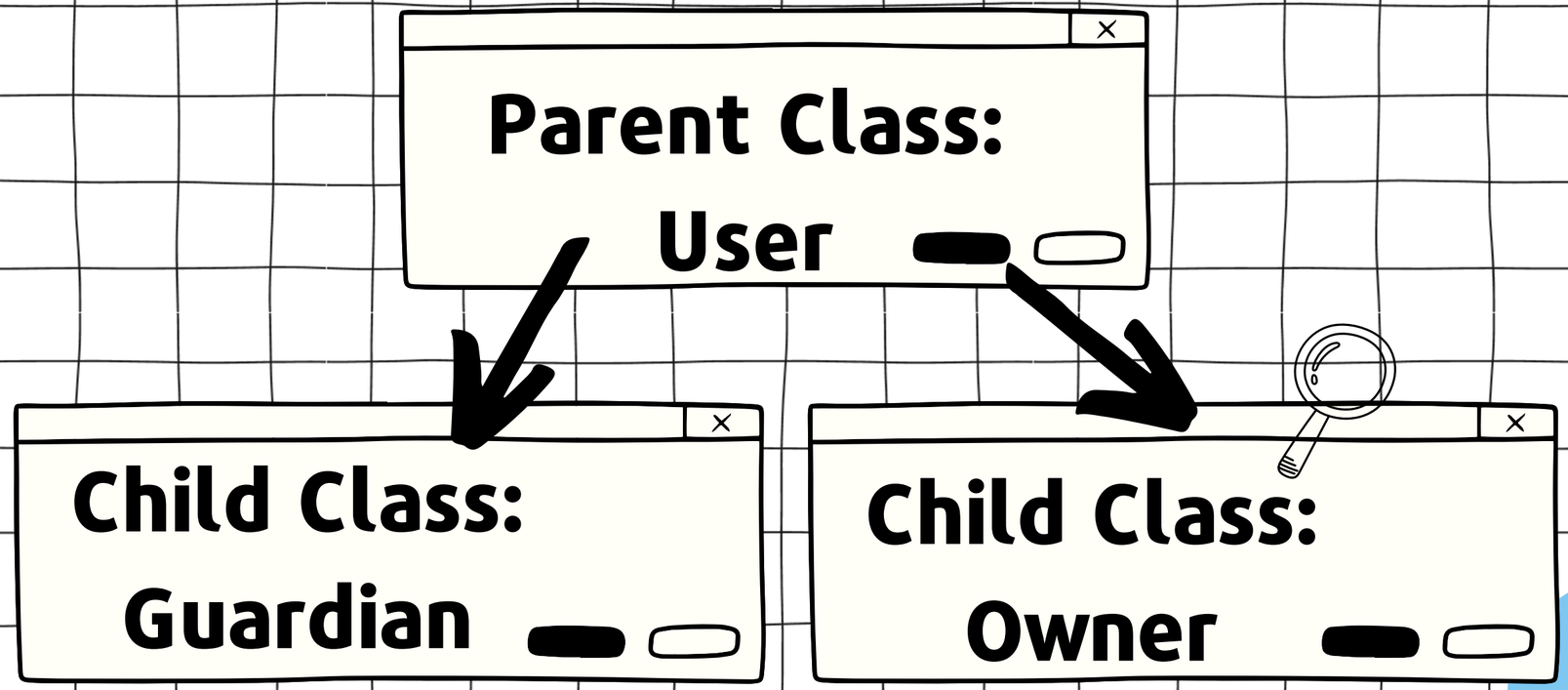


(object Dosage will be destroyed if object Medicine is destroyed, cannot exist independently)

Inheritance and Polymorphism

Inheritance

Inheritance provides a way to create a new class from an existing class.The new class is a specialized version of the existing class.The purpose of inheritance is generalization (sharing commonality between 2 or more classes) and specialization (extending the functionality of an existing class).



Polymorphism

Polymorphism is the objects perform the same action differently

Exception Handling

- In C++, exception typically means unexpected abnormalities that a program comes across during its execution. The process of handling this exception is called exceptions handling.

```cpp
#include <iostream>
using namespace std;


int main()
{
    int x = 6;

    // try block
    try {
        cout << "This is try block.";
        if (x > 0) {
            // throwing an exception
            throw x;
            cout << "After throw (Never executed) \n";
        }
    }

    // catch block
    catch (int x) {
        cout << "Exception Caught \n";
    }

    cout << "After catch (Will be executed) \n";
    return 0;
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\ACER9\OneDrive\Documents\SEMESTER 2\Programming Technique II\sem2 pt2> cd "c:\Users\ACER9\OneDrive\D
\sem2 pt2\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
This is try block.Exception Caught
After catch (Will be executed)
PS C:\Users\ACER9\OneDrive\Documents\SEMESTER 2\Programming Technique II\sem2 pt2>
```

Juliana Silva

Thank you