MedSchedule

Reliable reminders for your medicine

University of Miami
Department of Computer Science

CSC431 Group 1

Isaac Attuah
Grace (Tianjiao) Gu
Jin Curia

Table of Contents
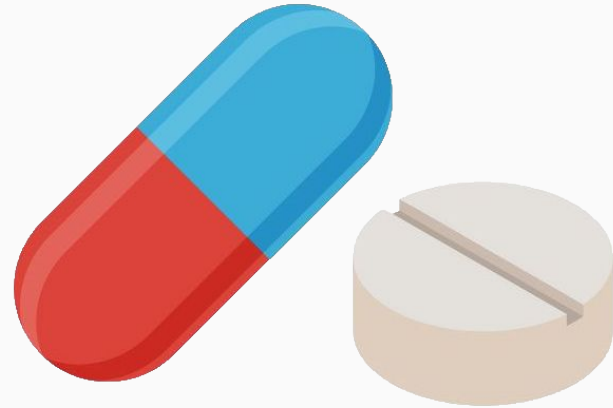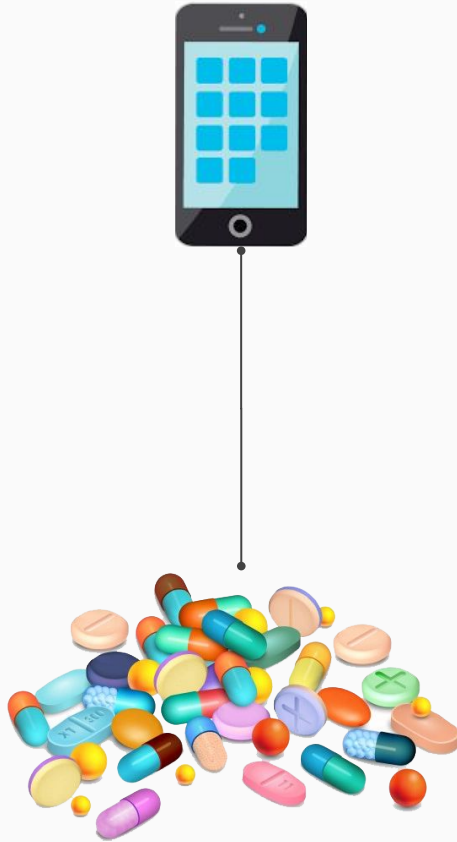
Background

People over age 65
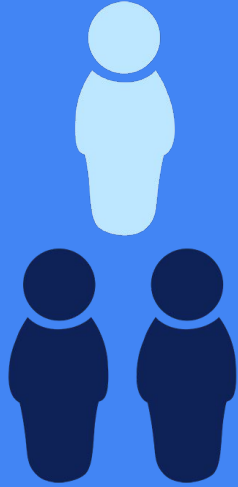
1 out of **3** people take

at least **5** different prescription medications

And **1** out of **4** people take

at least **10** different prescription medications

MedSchedule & Proposal

The purpose of this project is to create a mobile app for tracking medication doses, dose schedules, reminders, and refills.

The primary goal of this application is to simplify the process of organizing these health tasks by automation.
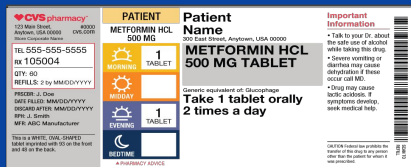
Functional Requirements
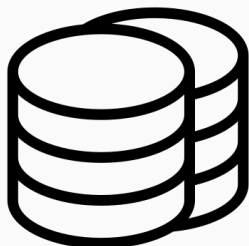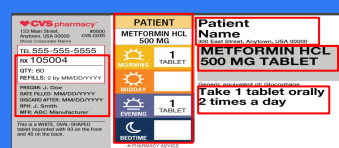
Register Medication

1. Medication Positioned

2. Camera Reads Input

3. Segmentation of Label

4. Segmentation of Information

Sign up and logging in

Start by creating an account.

Track your Medications starting today!

Create an account

Sign In

✕

Forgot password?

Enter your email below and we will send you a reset email.

Email
michael@gmail.com

Submit

Q W E R T Y U I O P
A S D F G H J K L
⬆ Z X C V B N M ⌫
123   space   Go

✕

Create a new password

Choose a strong password.

Password
●●●●●●●●●●●● 👁

Repeat Password
●●●●●●●●●●●● 👁

Sign in

Q W E R T Y U I O P
A S D F G H J K L
⬆ Z X C V B N M ⌫
123   space   Go

View medication information

Logged in → View Information
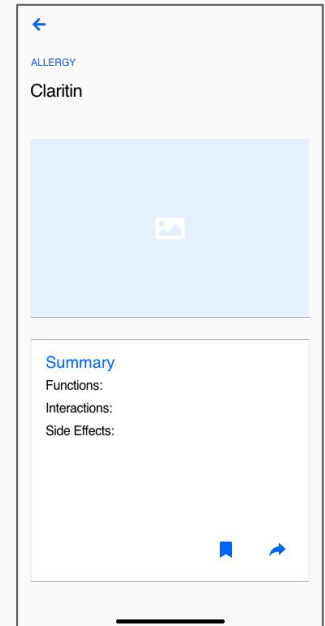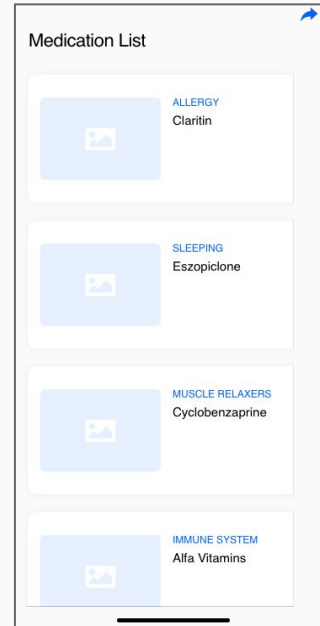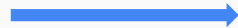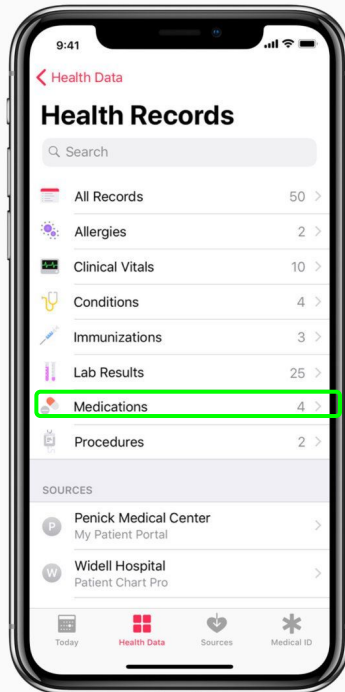
Set alerts
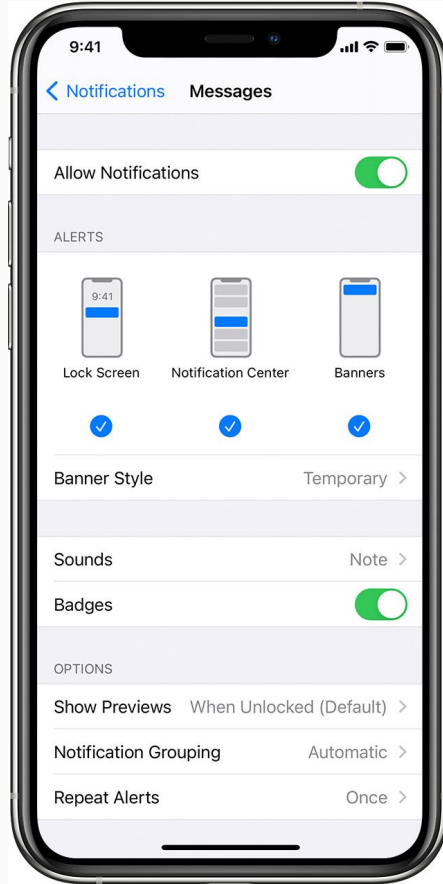
Settings ➡ Set alerts

Non-Functional Requirements

Non-functional Requirements

Secure Patient Information

Fast Image Segmentation

Alert Patients Without Delays

Enable Cross-platform Usage

System Constraints

Developer Expertise

Effort in
iOS development

Constant Updates
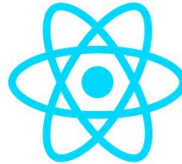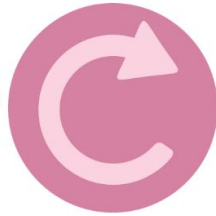
Size of app

Loosely typed language

Reliance on third parties

Use Case Diagram

UML Class Diagram

Account

- account_ID: int
- account_password: string
- is_authenticated: bool

+ account(self): Account
- authenticate_user(): bool

Schedule

- calendar_obj: Calendar

+ Schedule(): Schedule
+ updateCalendar(PatientAccount) : void

1..1

PatientAccount

- user_Dash: Dashboard
- user_settings: Settings
- user_Medications:
LinkedList<Medication()>
- user_Stock:
tuple(Medication*, int)
- user_Schedule: Schedule

+ PatientAccount(): PatientAccount

+ getMedications():
LinkedList<Medication()>

+setMedications(Medication*,int):bool

+ setUser_stock(Medication*, int):
bool
+ updateSchedule(): void
+ updateSettings(): void

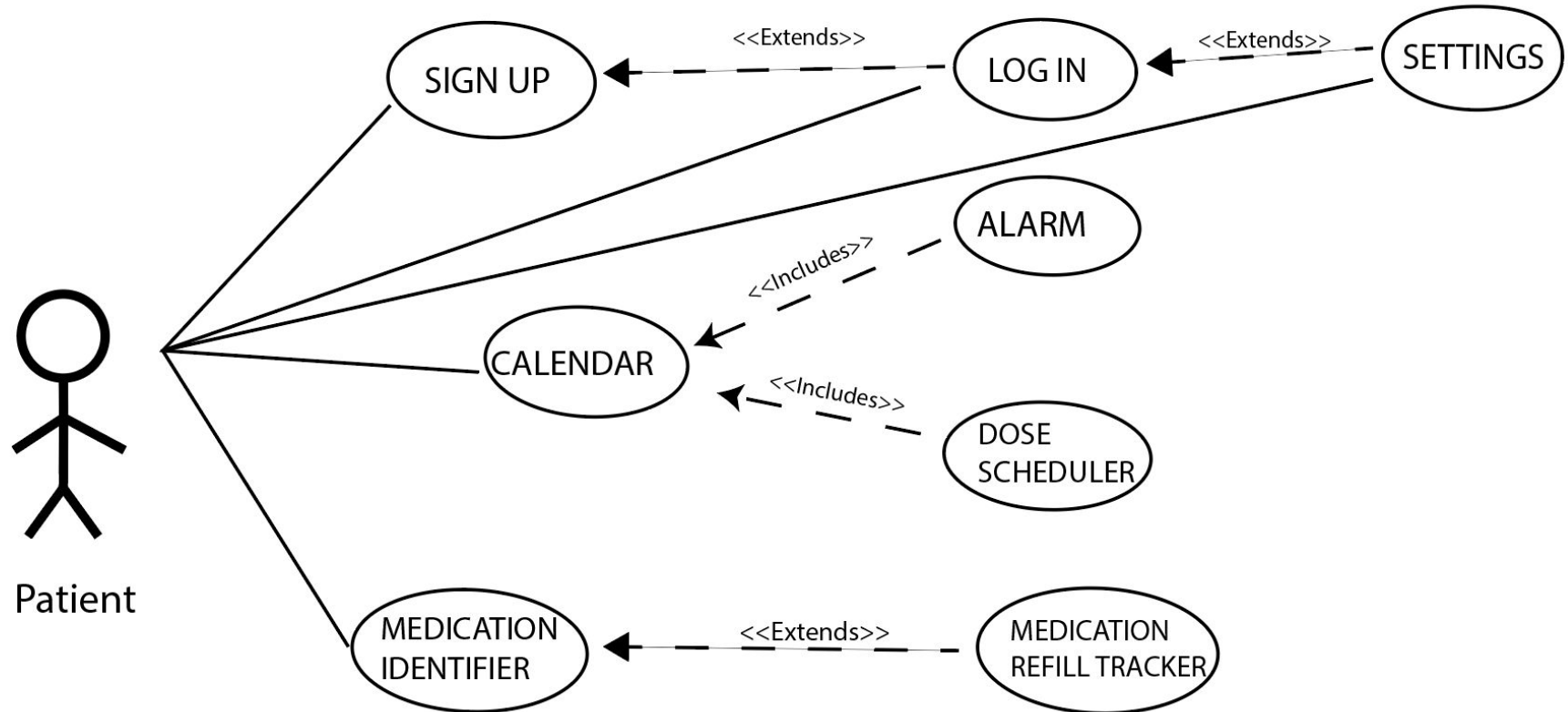1..1

1..1

Settings

- user_name : string
- user_age : int
- user_phone : string
- user_email: string

- PatientAccount():
PatientAccount

+ setPassword(string):
void
+ setName(string): bool
+ setPhone(string): bool
+ setEmail(string): bool
+ getName(): string
+ getPhone(): string
+ getEmail(): string

1..1

Counter

- quantity: int
- med_names: string[]
- results : string

- Counter(result): Counter
- setQuantity(Medication*,
int): bool
- setNames(Medication*,
int): bool
+
getQuantity(Medication*):
int
+ getResults(): string

1..*

..

Identifier

- counter_obj : Counter
- raw_med_list :
LinkedList<Medication>

+ Identifier()
- openCamera(): void
- useCamera(): void
- medicationDetected():
bool
- setRawMeds(Medication
*): void

1..1

1..1

Medication

- chem_name: string
- generic_name: string
- drug_interactions:
LinkedList<Medication*>

- Medication(): Medication
- setDrugInteractions(int,
Medication*): bool
- setChemName(string):
bool