



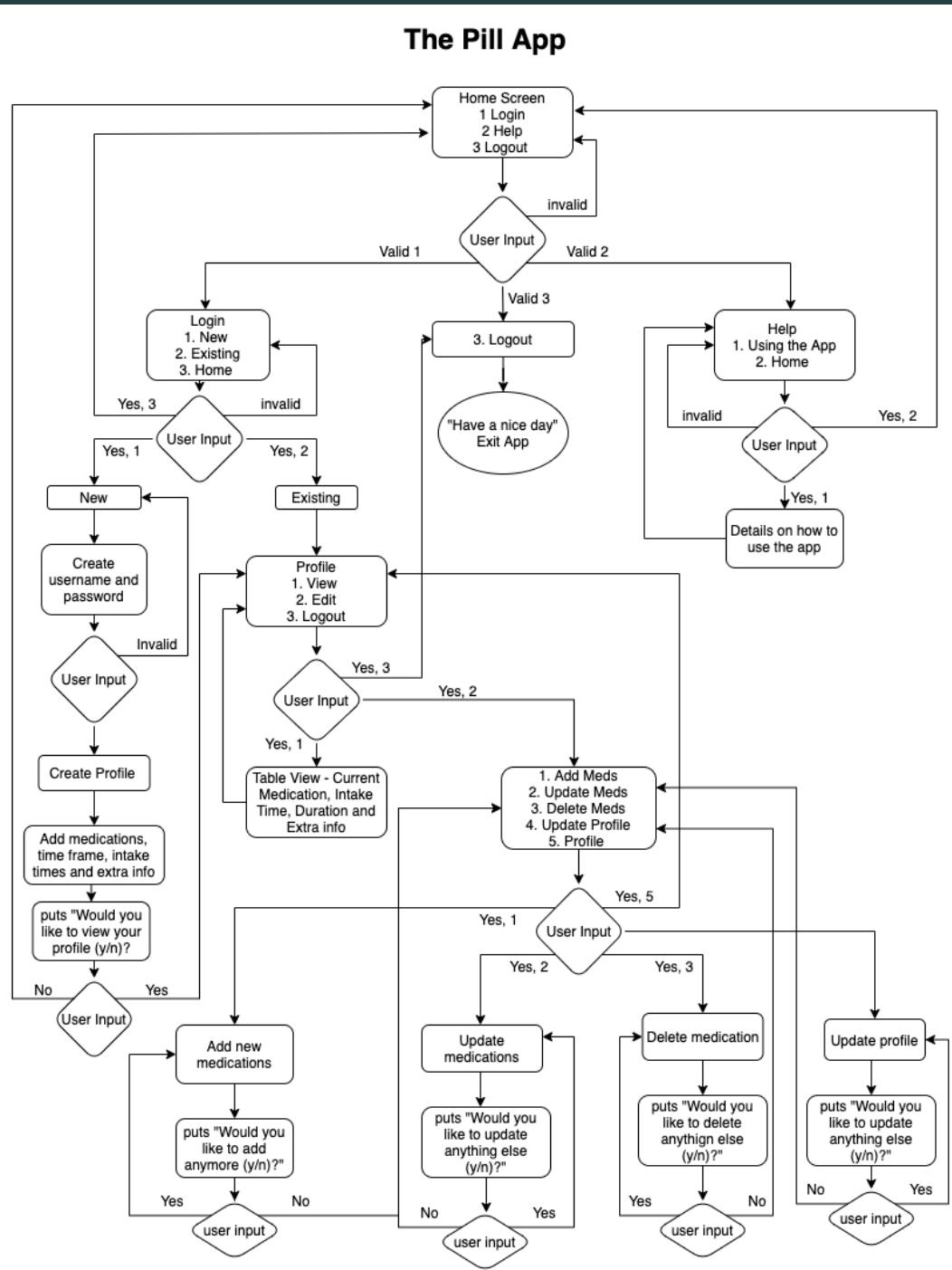
Terminal App



Problem

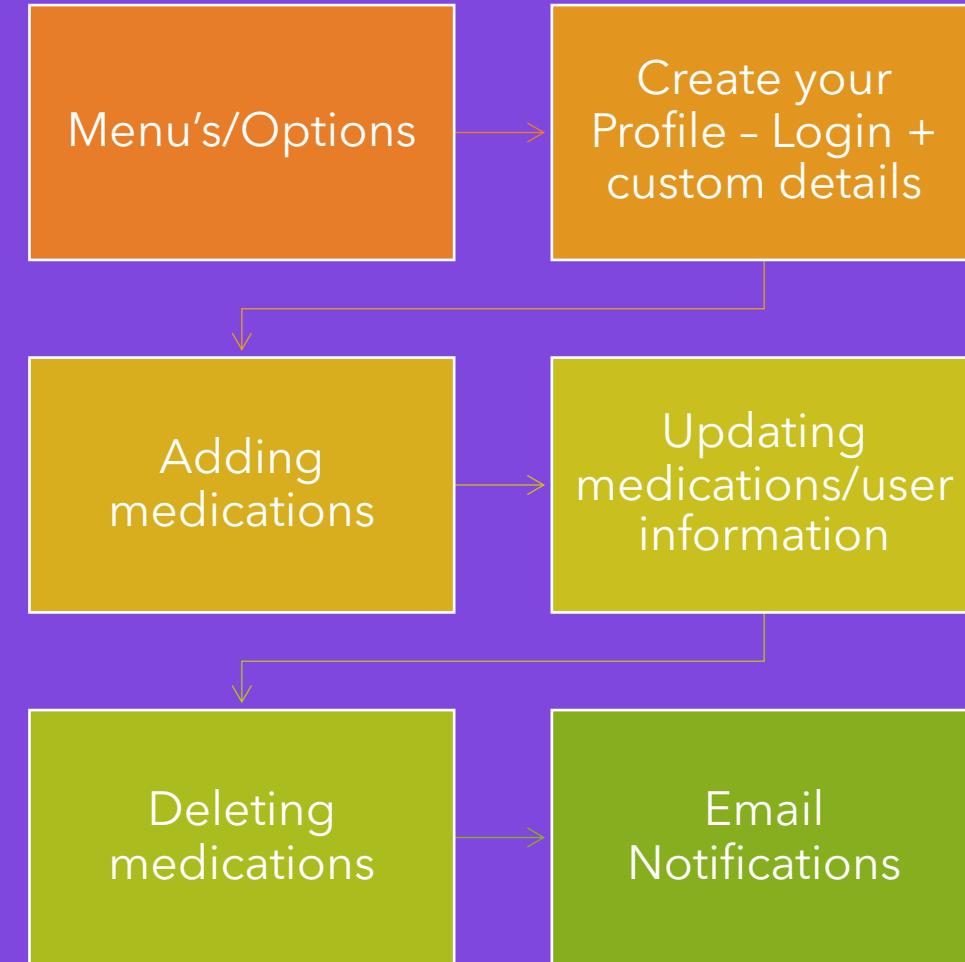
Solution

- An application where you can create a custom profile for yourself
- Enter the medication you'd need to take, the intake times and duration and whether it needs to be consumed before/after food.
- The application will then send you an email reminder
- The application will allow adding more medication, removing/updating medicine, deleting medications as well as updating user information such as email



Control Flow

Features



Menu

```
prompt = TTY::Prompt.new
option = prompt.select("Welcome to the Pill Box!".colorize(:red)) do |menu|
  menu.enum "."
  if login == false
    menu.choice "Login", 1
  end
  menu.choice "Help", 2
  if login == true
    menu.choice "Profile", 3
  end
  if login == true
    menu.choice "Logout", 4
  end
end
```

Create your profile

- Create username
- Create password
- Add in medication details

```
puts "Create your username:"  
name = user_input  
puts "Create your password:"  
password = user_input  
puts "Please enter your email:"  
email = user_input  
system("clear")  
user = {name: name, password: password, email: email}  
users << user  
write_user(users)  
login = true  
if login == true  
    render_logo  
    puts "Welcome #{name}, Let's create your profile"  
    puts "What medications would you like to add?"  
    meds = user_input  
    puts "When do you need to take this (e.g. Morning, Afternoon, Night, 2 times a day)?"  
    time = user_input  
    puts "How long do you need to take this for (e.g. 12 months, 6 months, 24 months)?"  
    duration = user_input  
    puts "When should you take the medication (options: before food, after food)"  
    info = user_input  
    system("clear")  
    profile = {Medication: meds, Intake_Time: time, Duration: duration, Extra_Info: info}  
    profiles << profile
```

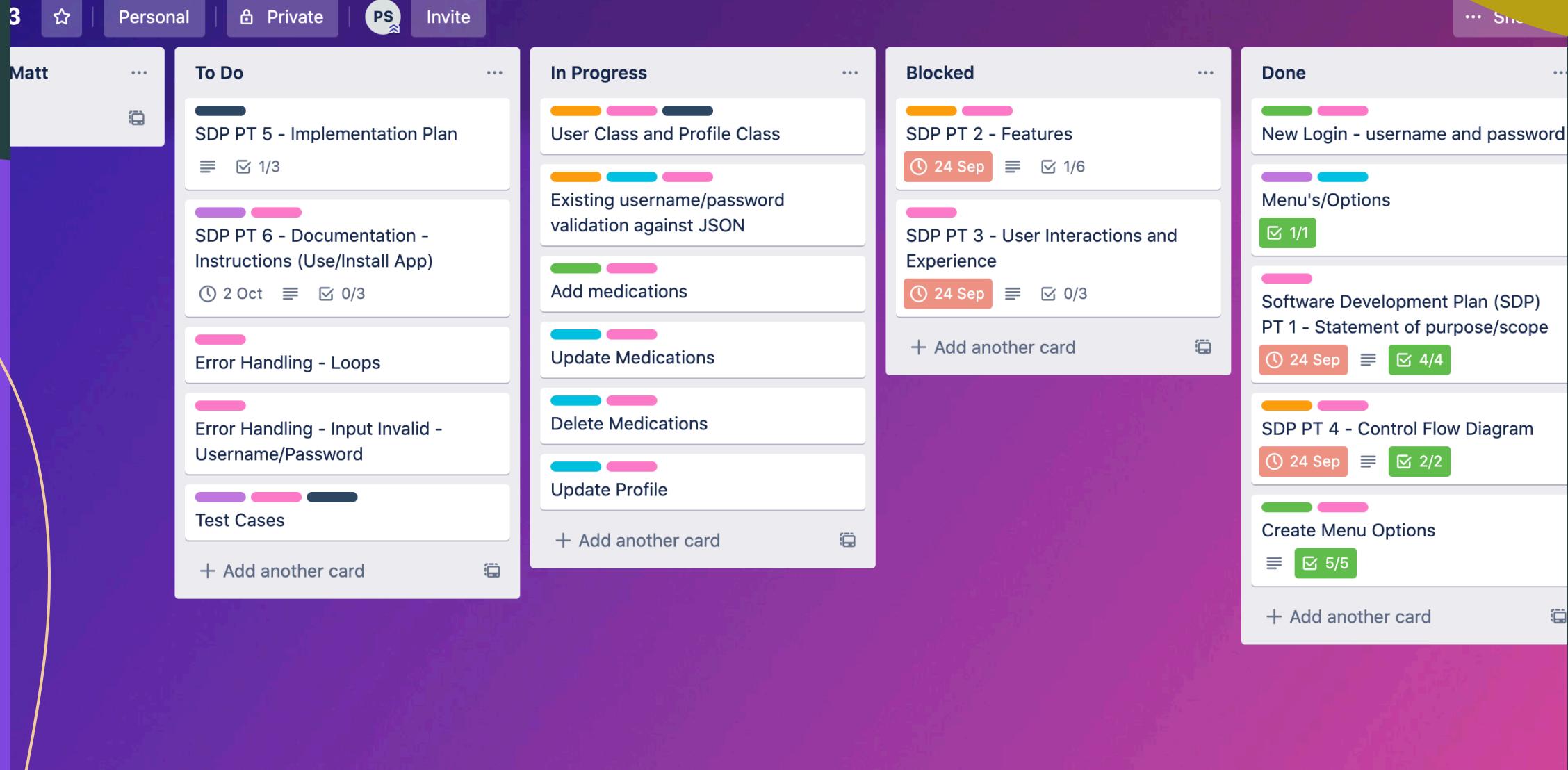
Gems

- Colorize
- Artsii
- Terminal Table
- TTY Prompt
- Pony

Files

JSON to capture and store data

```
def write_user(users)
  File.open("files/user_info.json", "a") do |f|
    f.write(users.to_json)
  end
end
```





Walk through

Future

- Existing linked to validate
- Update/Delete Meds
- Error Handling

Challenges

- Where to start
- Methods/Classes before code
- JSON

Favourite

- Gems
- Things working