

Com S 402 - Group SD10 - Demo 1

Physical Activity Self-efficacy Intervention Application

Jacob Kruse, Philip King, Jake Sweeney, Oscar Lenkaitis, Nolan Brown

The Problem

Issue:

Reports suggest adults with obesity do not meet public health guidelines for physical activity.

Intervention Strategy:

Evidence demonstrates that regular physical activity reduces the risk or progression of chronic diseases for this population, even without weight loss

Solution:

Develop a mobile app for a fitness research study that delivers curated fitness advice and resources to the test group while restricting access for the control group. The app will also automate email notifications to guide participants through study phases.

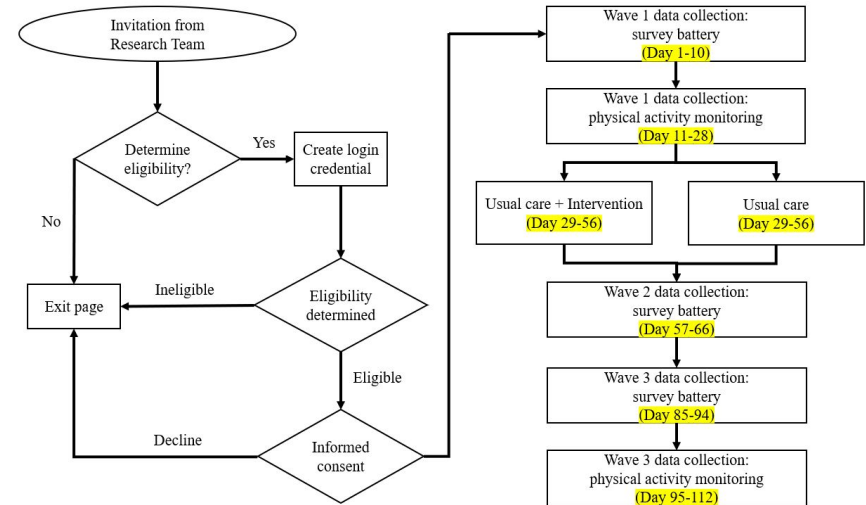


Com S 402 - Group SD10

Research Study Flow

Study Overview:

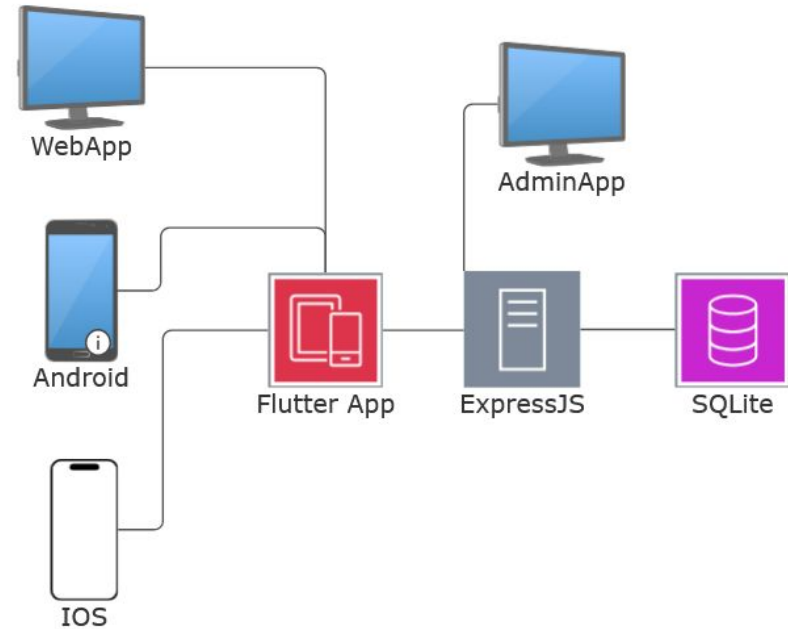
- **Duration:** 112-day research study using the PAS Mobile App
- **Eligibility:** Determined by specified user data
- **Groups:**
 - **Control Group:** No access to intervention materials
 - **Study Group:** Access to intervention videos and resources
- **Participation:** Users log in daily for updates and study materials



Com S 402 - Group SD10

Overall Design and Development

- GitLab ticket and milestones
- Figma UI Design
- Flutter Frontend Application
 - Dart
 - Secondary admin application
- Express/Node Js Backend
 - SQLite database
 - swagger
- AWS server for hosting
- Deploy PAS Application to the IOS and Google Play Store



Com S 402 - Group SD10

Frontend

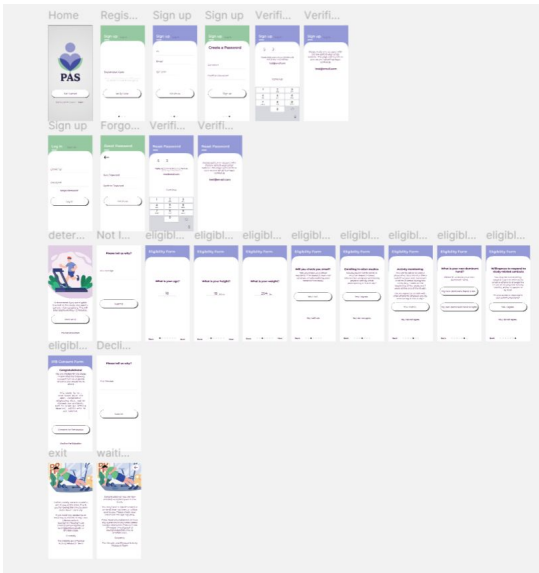
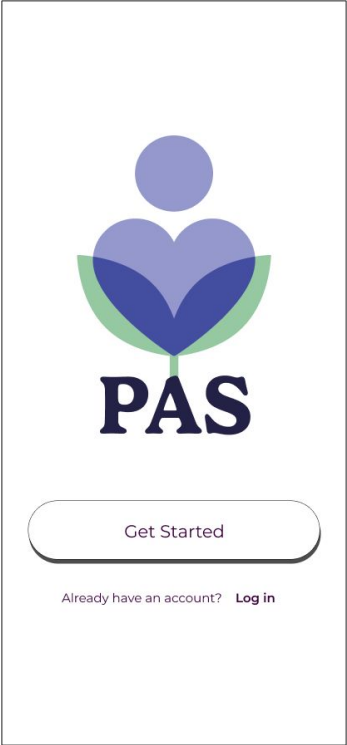
Com S 402 - Group SD10

Development Practices and Tools

- Using Figma to lay out our design as described in documentation

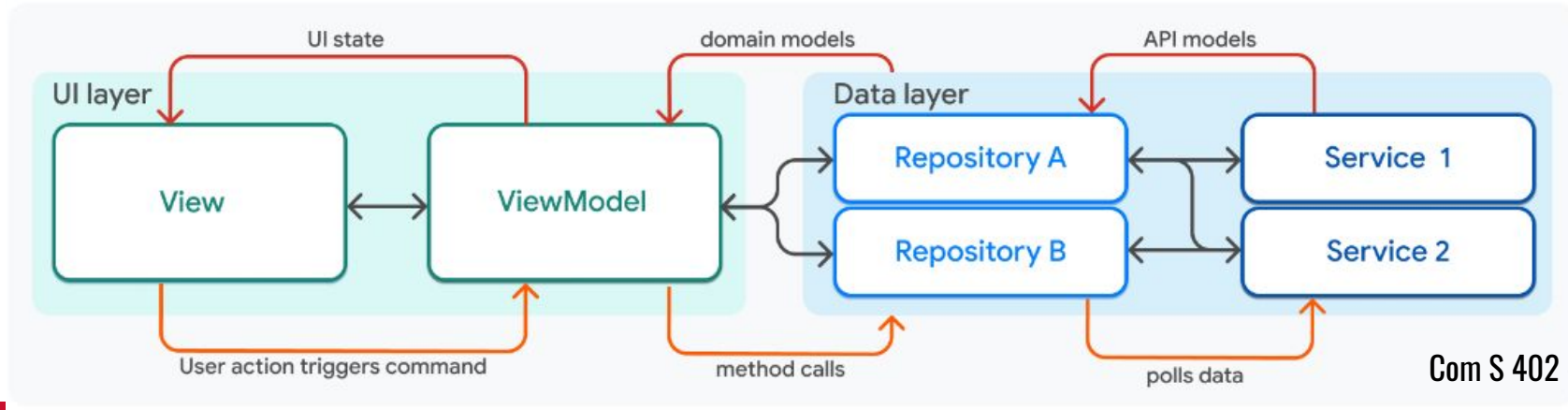
Table of Contents

INFORMATION 1: (APP) APP INFORMATION.....	3
INFORMATION 2: (APP) CREATE YOUR ACCOUNT	4
INFORMATION 3: (EMAIL) CONFIRM TO ACTIVATE YOUR ACCOUNT.....	5
INFORMATION 4: (APP) INTERESTED IN DETERMINING ELIGIBILITY.....	6
INFORMATION 5: (APP) ELIGIBILITY CRITERIA	7
INFORMATION 6: (APP) IRB CONSENT FORM	10
INFORMATION 7: (APP) EXIT SCREEN	11
INFORMATION 8: (APP) WAITING SCREEN.....	12
INFORMATION 9: (EMAIL) WAVE 1 ONLINE SURVEY SET – READY.....	13
INFORMATION 10: (EMAIL) WAVE 1 PHYSICAL ACTIVITY MONITORING – READY	14
INFORMATION 11: (APP) ENTER THE WAVE 1 PHYSICAL ACTIVITY CODE	15
INFORMATION 12: (EMAIL) PHYSICAL ACTIVITY MONITORING TOMORROW (WAVE 1)	16
INFORMATION 13: (EMAIL) SURVEY BY TODAY & RETURN MONITOR (WAVE 1).....	17
INFORMATION 14: (EMAIL) MISSING CODE ENTRY (WAVE 1).....	18
INFORMATION 15: (APP) DOUBLE-BLIND RANDOMIZATION	19
INFORMATION 16: (EMAIL) INTERVENTION ACCESS LATER.....	20
INFORMATION 17: (EMAIL) INTERVENTION ACCESS IMMEDIATELY	21
INFORMATION 18: (EMAIL) WAVE 2 ONLINE SURVEY SET – READY.....	22
INFORMATION 19: (EMAIL) NO WAVE 2 PHYSICAL ACTIVITY MONITORING	23
INFORMATION 20: (EMAIL) WAVE 3 ONLINE SURVEY SET – READY.....	24
INFORMATION 21: (EMAIL) WAVE 3 PHYSICAL ACTIVITY MONITORING – READY	25
INFORMATION 22: (APP) ENTER THE WAVE 3 PHYSICAL ACTIVITY CODE	26
INFORMATION 23: (EMAIL) PHYSICAL ACTIVITY MONITORING TOMORROW (WAVE 3)	27
INFORMATION 24: (EMAIL) SURVEY BY TODAY & RETURN MONITOR (STUDY END).....	28
INFORMATION 25: (EMAIL) MISSING CODE ENTRY (STUDY END).....	29
INFORMATION 26: COMPRESSED SCHEDULE & DATA DOWNLOAD	30



Development Practices and Tools

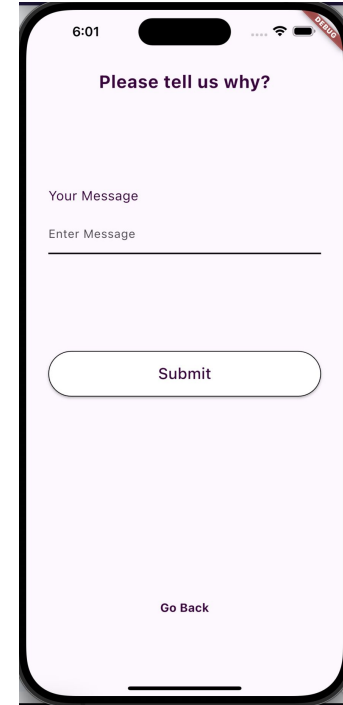
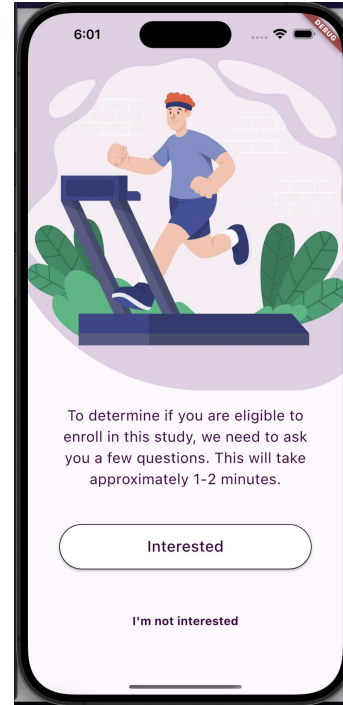
- Using flutter to develop both iOS and Android consecutively
- MVVM Architecture
 - View holds display widgets
 - Viewmodel holds state, manages data
 - Majority of logic is in ViewModels
- Data layer connects to backend services
 - Makes api calls
 - Returns data to UI views
 - Handles caching, data refresh, system calls, etc.



Com S 402 - Group SD10

Current Progress Status

- Frontend architecture
- Custom button utility
- Custom view transition
 - Transition speed
 - Direction
 - Animation easing
- Current views in progress
 - Home
 - Login
 - Eligibility



Com S 402 - Group SD10

Current and Upcoming Challenges

- Learning dart/flutter
- Frontend/backend communication
 - API calls for user data
- Embedded Vimeo video player
 - Mini games
- Future sprint planning
 - Main content views

The image displays a sequence of eight mobile app screens for an eligibility form. Each screen has a purple header with the text "Eligibility Form" and a progress indicator at the bottom consisting of a row of seven dots, with the current step's dot filled.

- Screen 1:** "What is your age?" with a text input field containing "18".
- Screen 2:** "What is your height?" with a text input field containing "18 inches".
- Screen 3:** "What is your weight?" with a text input field containing "264 lbs".
- Screen 4:** "Will you check you email?" with a sub-question: "Will you check your email regularly? The research team will send you emails outlining your tasks for this study." Two buttons: "Yes, I will." and "No, I will not.".
- Screen 5:** "Enrolling in other studies" with a sub-question: "Do you agree not to enroll in another research-based intervention program promoting physical activity while participating in this study?" Two buttons: "Yes, I agree." and "No, I do not agree.".
- Screen 6:** "Activity monitoring" with a sub-question: "You will be asked to wear a physical activity monitor (like a watch) on your non-dominant wrist for 2 weeks during this study (e.g., 1 week at the beginning of the study and 1 week at the end of the study). Do you agree to comply with instructions for physical activity monitoring in this study?" Two buttons: "Yes, I agree." and "No, I do not agree.".
- Screen 7:** "What is your non-dominant hand?" with a sub-question: "Please let us know your non-dominant hand." Two buttons: "My non-dominant hand is left." and "My non-dominant hand is right.".
- Screen 8:** "Willingness to respond to study-related contacts" with a sub-question: "You may be contacted by research team members via email or phone to arrange the return of the physical activity monitor, either in person or by mail. Do you agree to respond to our communications?" Two buttons: "Yes, I agree." and "No, I do not agree.".

Com S 402 - Group SD10

Backend

Com S 402 - Group SD10

SQLite and Rest API

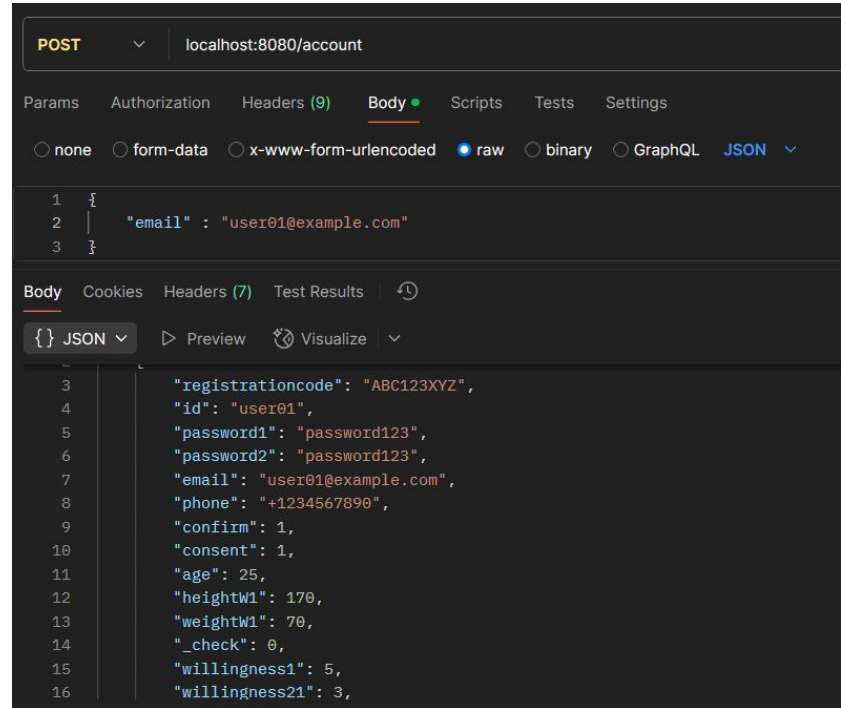
- Research participants need to have an account
 - Personal information
 - Login information
 - Study status
- Frontend needs to update database
 - Create account
 - Login
 - Check status
 - Admin interface
- Node.js was chosen to handle API and database, as well as email and scheduled tasks
 - Simple and easier to hand off
- Security is important since personal information
 - Access control in the form of authentication token



Com S 402 - Group SD10

Current Progress Status

- Current Status
 - SQLite account table and access functions
 - Express API, account retrieval
- Challenges
 - Email verification coordinating with frontend and handling account state
 - System for managing intervention content
 - Set up AWS for official deployment



```
POST localhost:8080/account

Params Authorization Headers (9) Body Scripts Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   "email" : "user01@example.com"
3 }
```

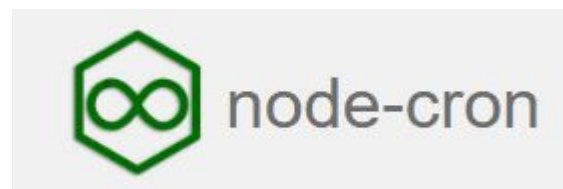
```
Body Cookies Headers (7) Test Results
{} JSON Preview Visualize

3   "registrationcode": "ABC123XYZ",
4   "id": "user01",
5   "password1": "password123",
6   "password2": "password123",
7   "email": "user01@example.com",
8   "phone": "+1234567890",
9   "confirm": 1,
10  "consent": 1,
11  "age": 25,
12  "heightw1": 170,
13  "weightw1": 70,
14  "_check": 0,
15  "willingness1": 5,
16  "willingness2": 3,
```

Com S 402 - Group SD10

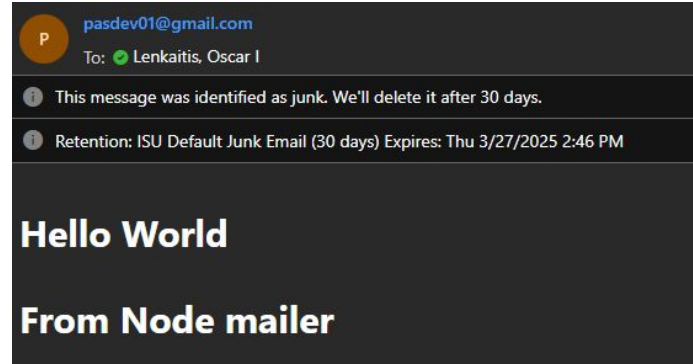
Emailing and Task Scheduling

- Emailing participants is a big portion of the backend requirements.
- Emailing tasks include
 - Email verification
 - Password resets
 - Notifying of next steps after established amount of time (days/weeks after previous tasks)
- Giving access to certain resources after established amount of time.
- Confirm tasks have been completed within a certain time frame.
- Using nodemailer to use SMTP from an admin email account.
- Using node-cron to run mailing functions and all other scheduled task.



Current Progress Status

- Can send a basic test email from the backend localhost.
- Current Challenges
 - Emails are not verified. Get sent to spam folder.
 - Emails are not secure/encrypted.
 - Emailing limits imposed by mailing protocols and free storage
- Have set up basic tasks to test and understand the API of node-cron.
- Current Challenges
 - Lack of test data.
 - Need to establish APIs and protocols



```
Cron initialized...  
Example app listening on port 443  
Email sent  
Ping at 1:46:30 PM  
Ping at 1:46:36 PM  
Ping at 1:46:42 PM  
Ping at 1:46:48 PM  
Ping at 1:46:54 PM  
Ping at 1:47:00 PM  
Ping at 1:47:06 PM  
Ping at 1:47:12 PM  
Ping at 1:47:18 PM
```

Com S 402 - Group SD10

Remaining Work

- Set up with AWS Simple Email Services (SES) to send emails securely
- Create email functions and formatting for all use cases
- Create password reset protocol and tokens
- Schedule automated tasks to verify participant information, monitor resource access, and track study progress for each individual.

Demo

Com S 402 - Group SD10

Thank You!

Com S 402 - Group SD10