

Screen Sketches

1_AN_4

Tianyi Luo

FitQuest

Actors and Functionality

The name of our project is FitQuest. The purpose of this application is to create a platform for people who are doing fitness exercises or are planning to control weight to log their exercises and set up goals. The actors of this application are user, system administrator, business administrator and users.

First actor is the system administrator. The administrator will be able to access and manage system files on the server to make patches. They would be able to log files and fix the data problems for system maintenance.

The second actor is the business administrator. The business administrator would be able to manage user accounts including adding and deleting users, assigning roles and permissions. They would also be able to set up terms and policies which users would be accepting when using this app. They would be able to provide technical support for the users.

The third actor would be the user. The user would be able to create an account using their email address. Login in system would remember the user for a period of time. After they have logged in, they will be able to edit their profile which includes their username, weight, age, gender and we will be able to collect the data. Setting up exercise goals will be another function the users can have goals for short and long term. Planning workouts function would make the daily workouts easier for the users, they can keep a record of what they are planning to lift. Recording exercise would let the users keep track of what they did and know if they are being delicate or not. Record diet function will help users to track the calorie and substance intake daily. We will calculate the calories the user should take in order to help reach their goal. Creating posts and add-and-chat with friends would create a social network so the users can enjoy their exercise.

Non-functional Requirements

- For adding/retrieving data like Workout stats or diet record, response time should be less than 1 seconds.
- Changing of page or redirection should also take less than 2 seconds.
- At least 10 users should be able to use the app at the same time with minimum time difference compared to when only one user is online.
- All code should be documented properly, a 5 year old should be able to understand.
- Code should be written with the addition of features and possibility of heavy modification in the future in mind.
- UI should be user friendly, i.e., it should be easy to use, all symbols should be appropriate and decipherable. No need for a booklet to understand what's going on
- Everything should look comfortable to use and easy on the eyes.
- App should be able to run on an internet browser in the future with little to no change required at the backend and database.

Tables

Users

1. userID - primary key
2. username
3. password
4. email
5. gender
6. age
7. weight - N/A while signing up, can be assigned on edit profile page

Workouts(All)

1. workoutID - primary key
2. workoutName
3. setsApplicable - 1 means yes, 0 means no, will be shown as checkbox on frontend
4. weightApplicable - 1 means yes, 0 means no, will be shown as checkbox on frontend

Workout(One)

1. dayID - primary key
2. dailyWorkoutID - foreign key
3. date - date of workout
4. workoutReps - always applicable, minimum 1 if workout done on given day
5. workoutSets - if applicable, else option won't be available
6. workoutWeight - if applicable, else option won't be available

DailyWorkout

1. dailyWorkoutID - primary key
2. Date - date of workout

Friends

1. friendshipID - primary key
2. userOneID - foreign key referencing userID from table Users of user who sent friend request
3. userTwoID - foreign key referencing userID from table Users of user who receives friend request
4. enabled - 0 means not accepted, 1 means accepted, 2 means rejected. Resets to 0 if rejected and requested again. Creates new entry if completely new request.

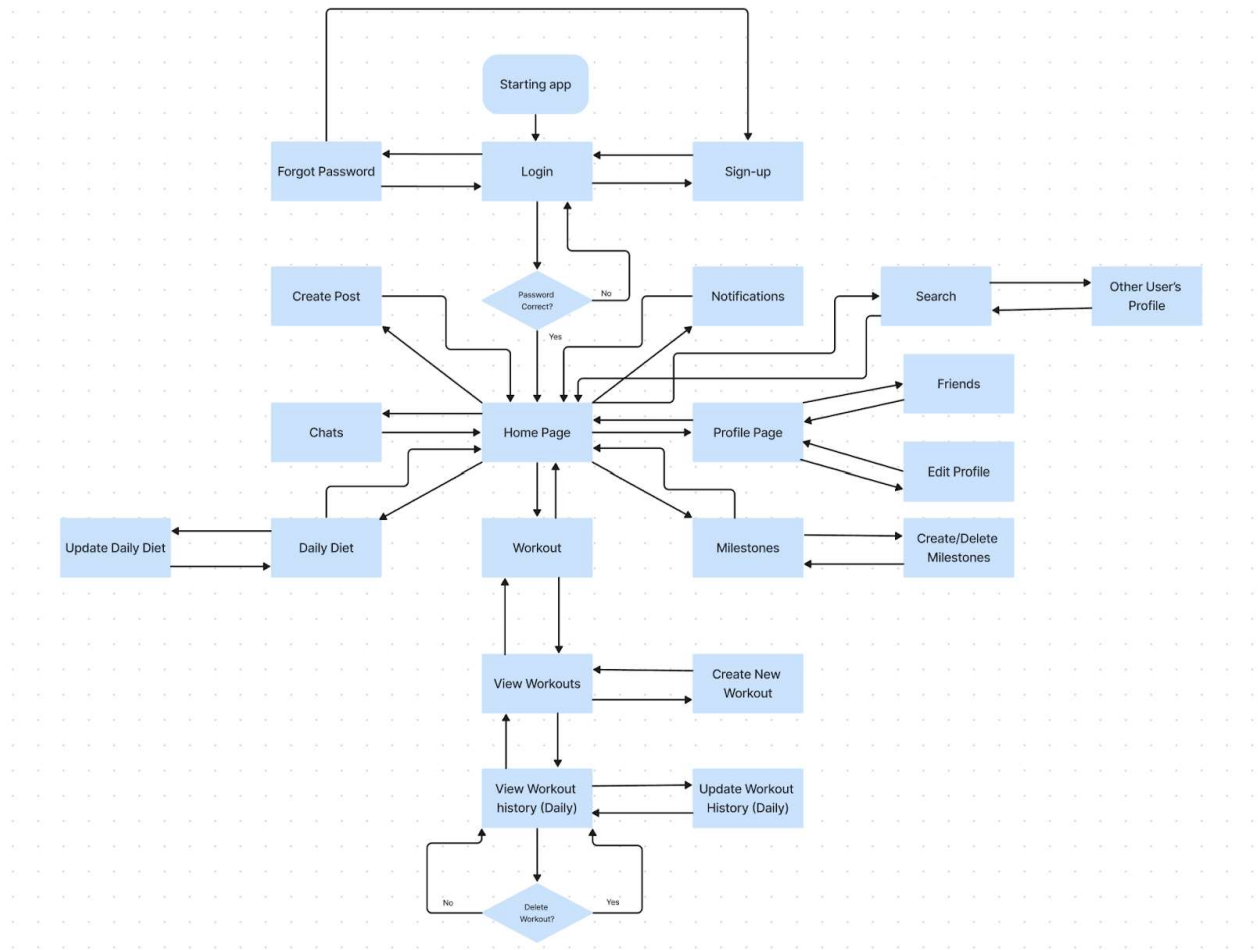
Diet

1. dietID - primary key
2. mealName
3. dietDate
4. dietCalories
5. mealNumber - corresponds to the time input on the add diet page

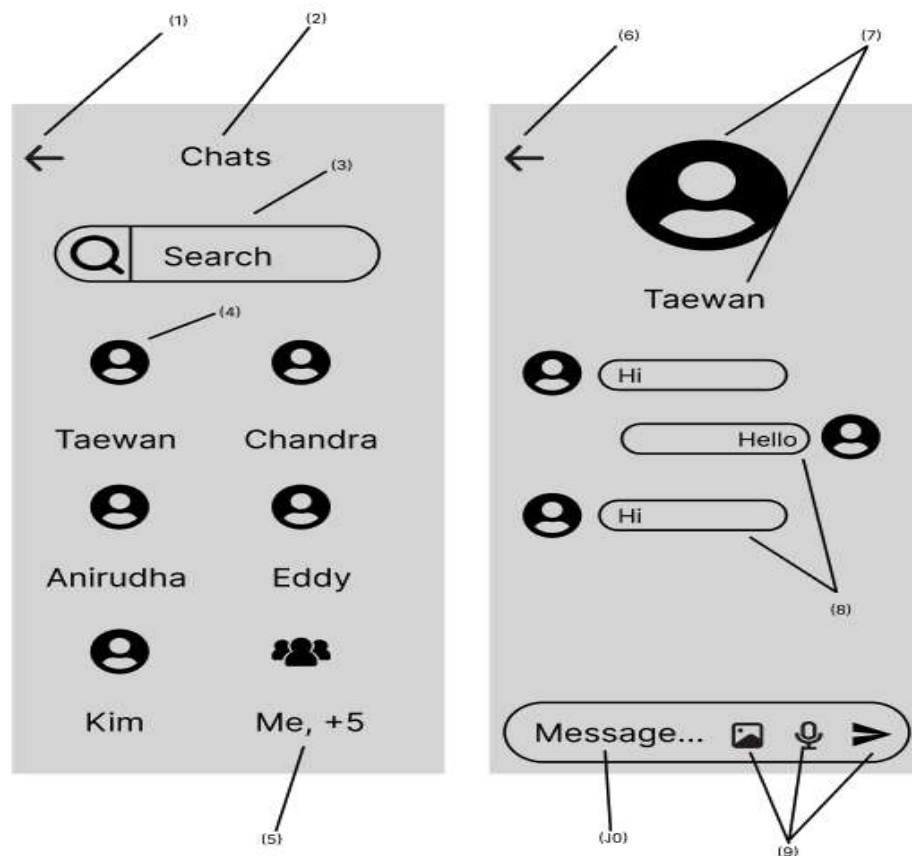
Chats

1. chatID - primary key
2. senderID - foreign key references userID of person sending the message
3. receiverID - foreign key references userID of person receiving the message
4. message
5. messageTime

Screen Flow Diagram



Chat



Purpose: To chat with users that are friends.

Loading event: This chat screen will pop up when the message icon from the main menu is clicked.

(1): This button will take the user back to the main screen.

(2): This shows that the user is on the chat screen.

(3): This is where the user will enter the username if they want to find a specific account to start to chat with.

(4): This is the user's icon and once clicked, it lets you to the chatting screen.

(5): This is showing that it's a group chat and once clicked, it lets you to the chatting screen.

(6): This button will take the user back to the chats screen.

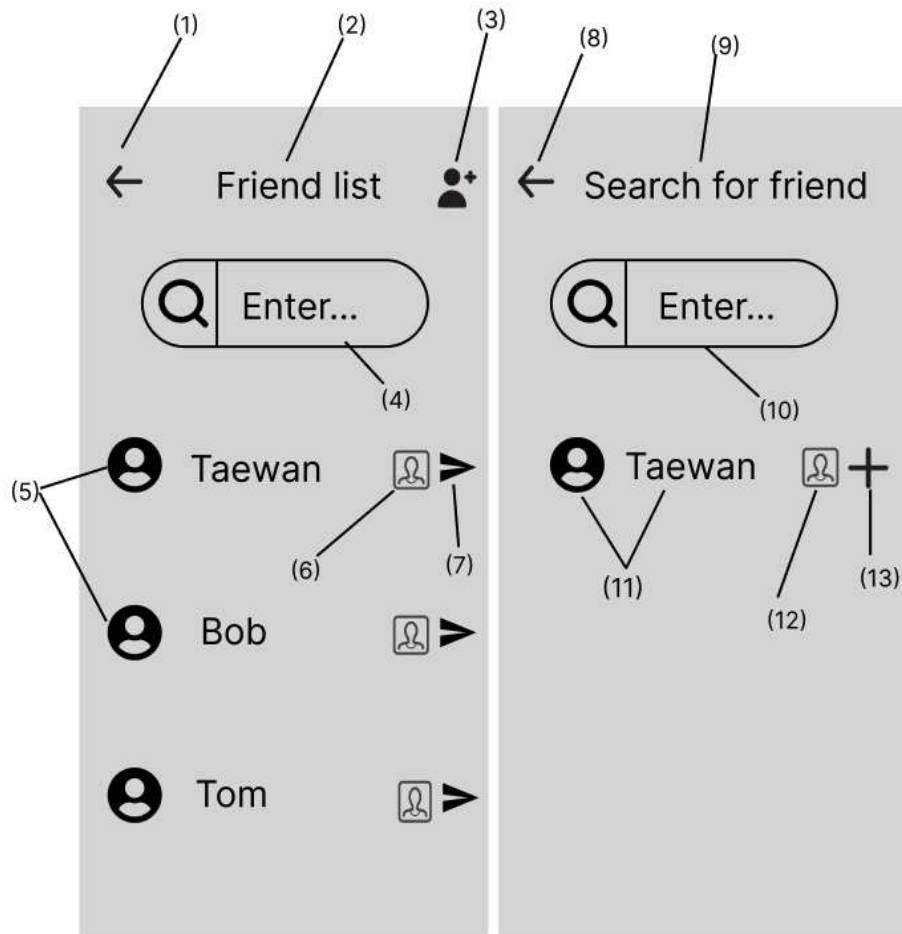
(7): This shows the user's profile picture and username.

(8): This shows the user's text

(9): These buttons are for sending a message, a photo, or a voice message.

(10): This is where the user will enter the text to send to the other user.

Add friend/friend list



Purpose: To add and see friend user

Loading event: This friend screen will pop up when the search icon is clicked on the main menu

(1): This button will take the user back to the main screen

(2): This shows that the user is on the friend list screen

(3): This button will take the user to the search for friend screen

(4): This is where the user will enter the username(email) to find a specific user to view and find friends.

(5): This shows the profile picture of the user

(6): This button will take the user to the profile page of the selected user

(7): This button will take the user to the chatting screen with the selected user

(8): This button will take the user back to the friend list screen

(9): This shows that the user is on the search for friend screen

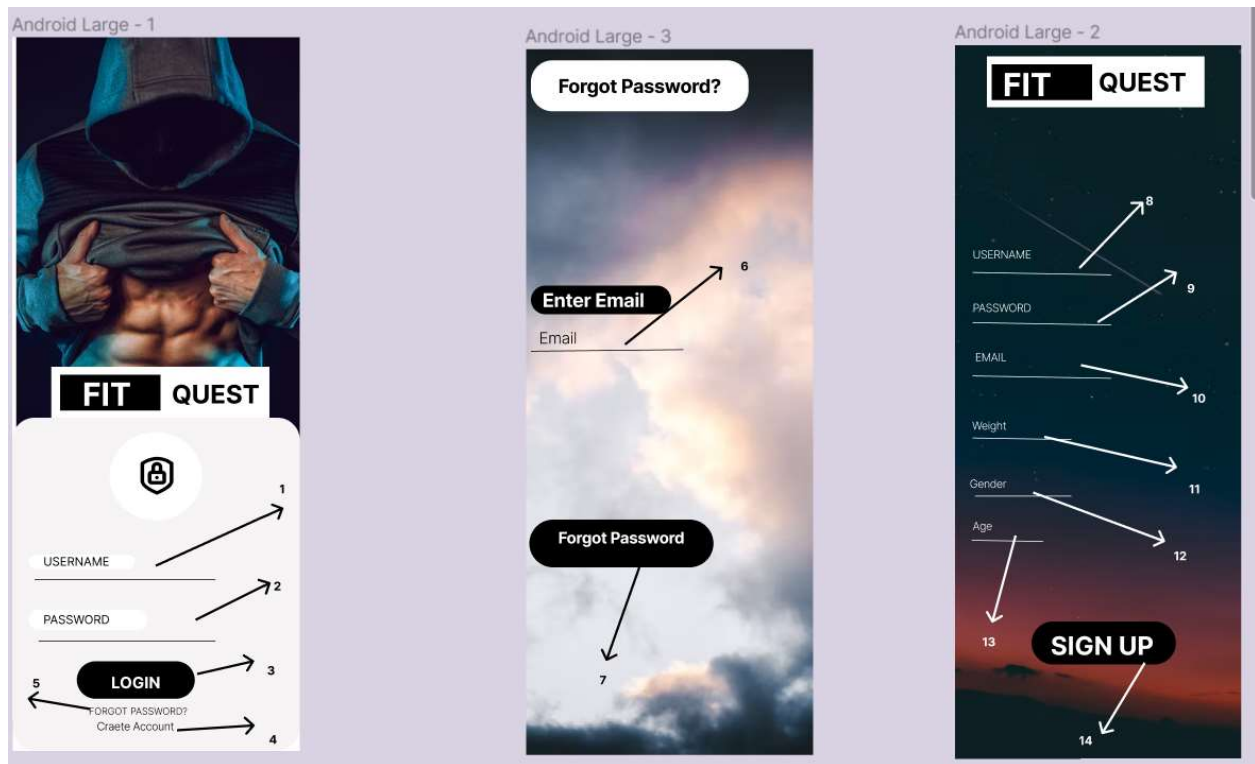
(10): This is where the user will enter the username(email) to find a specific user view and add a friend.

(11): This shows the username and profile picture of the user.

(12): This button will take the user to the profile page of the selected user

(13): This button allows the user to send a friend request to the selected user.

Login/Registration:



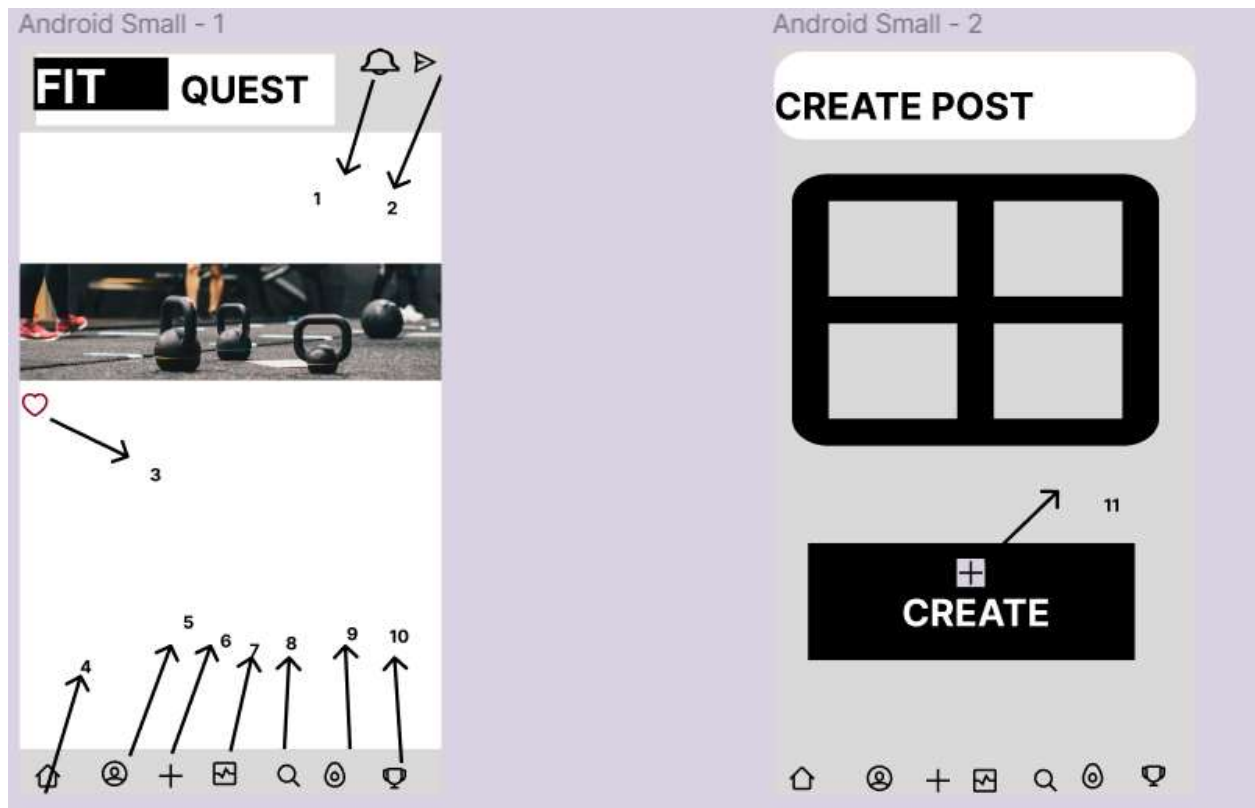
The main purpose of this page is for logging and registration:

It will have 3 pages : 1 for the login, 2 for the sign up and 3 for the forgot password.

- 1) The field USERNAME asks the user to enter his username which he gave while he registered .
- 2) It asks user to enter his registered password.
- 3) The Login button will verify the details and if everything is correct it prompts the user to the home screen.
- 4) If user didn't create account "Create account" field will prompt user to create his account before login.
- 5) If the user forgot his password they can click on "Forgot Password" this will ask the user for his registered email which acts as the path variable to get his password to his registered mail.
- 6) This is the field where the user can enter his registered mail to get his password if he forgot.
- 7) The forgot password button will send the user's password to his registered mail.

- 8) This is the field in the registration where the user creates their username and it will stored in the database.
- 9) This is the field in the registration where the user creates their password and it will stored in the database.
- 10) This is the field in the registration where the user enters their email and it will stored in the database.
- 11) 11,12,13 in this field the user enters their weight, gender and age and it will stored in the database.
- 12) Mentioned in 11
- 13) Mentioned in 11
- 14) The Sign Up button registers the user and they can login to enter the application. The data stored in the Sign Up page will be reflected in the user profile.

Home Screen and Create Post:

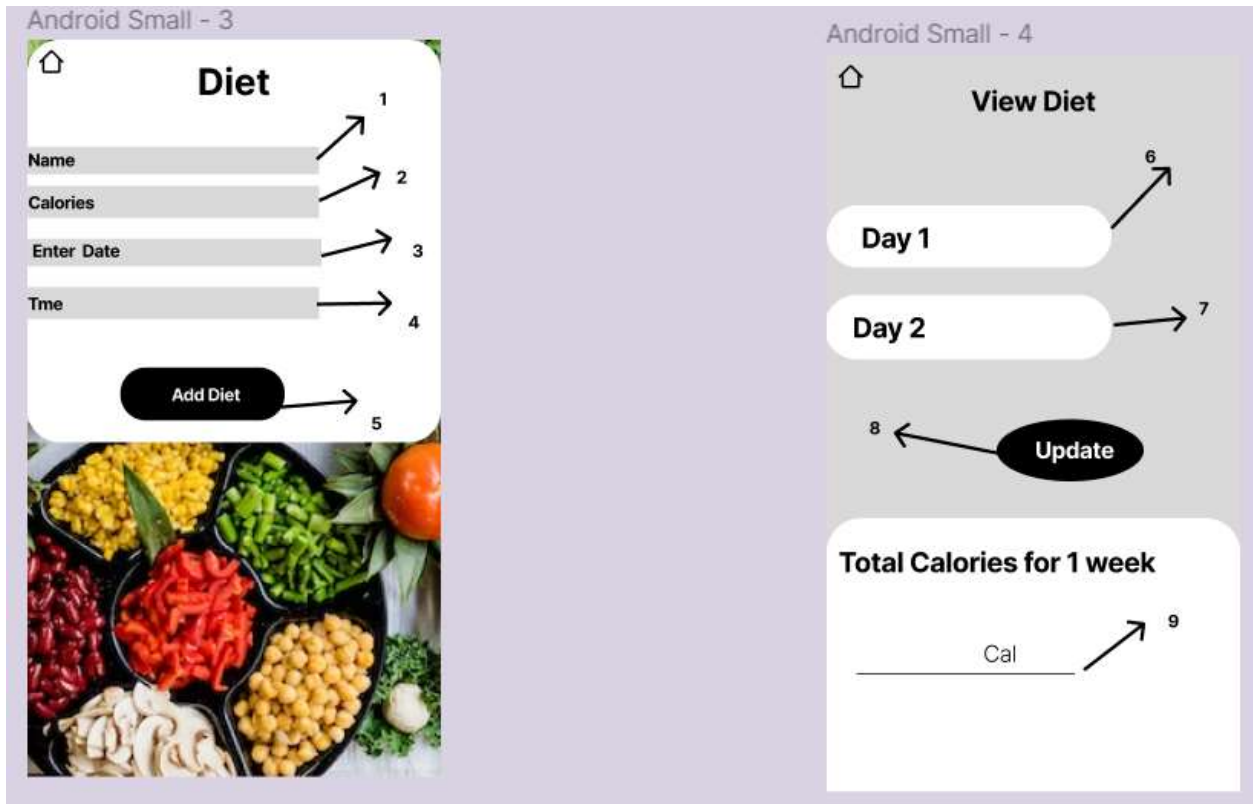


These two pages are the home page and create post page with a navigation bar at the bottom.

1. This button redirects the user to any notifications they might have received like friend requests.
2. This button opens the chats page where the user can see all the conversations they have with other users.
3. This is the like button for a post. The image is a placeholder and the posts will be related to the workouts and progress of the users. They will still have an option to add images if they want to. The posts are retrieved from the database after any user posts them.
4. This button redirects the user from any page back to the homepage.
5. This is the profile button. The user can access their profile page by using this button.
6. This button takes the user to the create posts page where the user can create posts. All posts are stored in the database
7. This button redirects the user to the workouts page. The user can view, create and delete workouts here.
8. This takes the user to a search page where the user can search for their friends and visit their profiles to add them as friends.
9. This button takes the user to the diet page. Here the user can track their diet, view how many calories they have eaten on average per day or their entire weekly calorie intake and more

10. This takes the user to the milestones page. The user can here view all the milestones they have set and all the milestones they have achieved.
11. This button creates a new post in the database. The user who posted and the other users who are friends of this user can then see this post in the home page which will retrieve the post from the database.

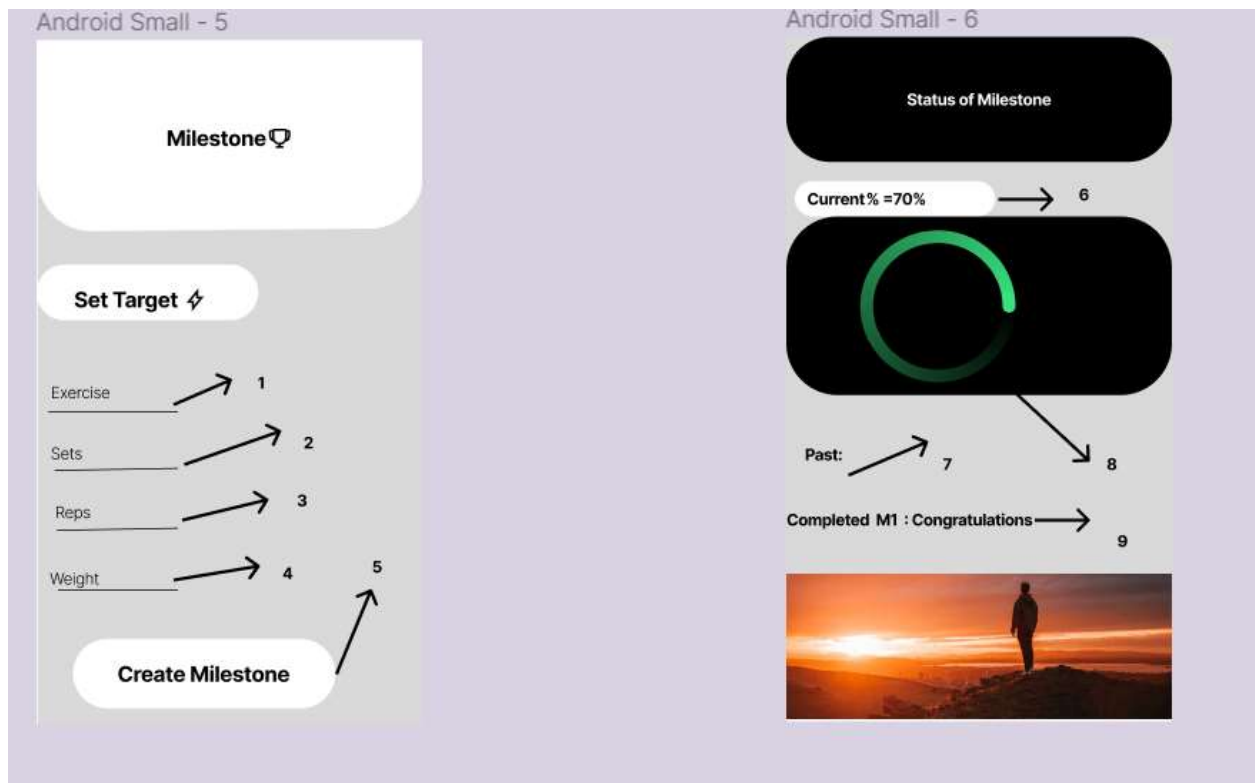
Diet and View Diet:



The main purpose of this screen is to track the diet component in our application..

- 1) This is the field where the user should enter the name of the food he wants to tack on.
- 2) They will then be asked to enter the calories of that food item.
- 3) They will then be asked to enter a date on when the user has had their meal. So this is the date which will act as the starting point to keep track of the diet data.
- 4) This will ask the user to enter the time on when he ate that food.
- 5) Add Diet button will add the diet to the View Diet page and the database.
- 6) This the diet or food tracking data in Day 1
- 7) This the diet or food tracking data in Day 2, as more are added, more will be visible
- 8) The Update button will allow users to update the the data of their diet.
- 9) This field shows the weekly calorie intake of the user.

Milestone/ Status of Milestone



1. This is the exercise the user wants to set a milestone for
2. This is the number of sets the user wants to set the milestone for said exercise
3. This is the number of reps the user wants to set the milestone for
4. This is the weight the user wants to set the milestone for
5. This button posts the milestone on the database and can be viewed in a page called "All milestones"
6. This is the progress on one selected milestone over the past week.
7. Closest sets, reps and weight user has done so far to achieving the milestone in the last 1 week. All the values will be displayed in one sentence. Eg:- "x sets, y reps and z kgs"
8. The shown image is just a placeholder for a progress bar. It visually represents the closest the person has gotten to achieving the milestone over the past week.
9. This shows status of the milestone. When achieved, will show a congratulations.

name: (1) name: (8)

age: (2) age: (9)

gender: (3) gender: (10)

weight: (4) weight: (11)

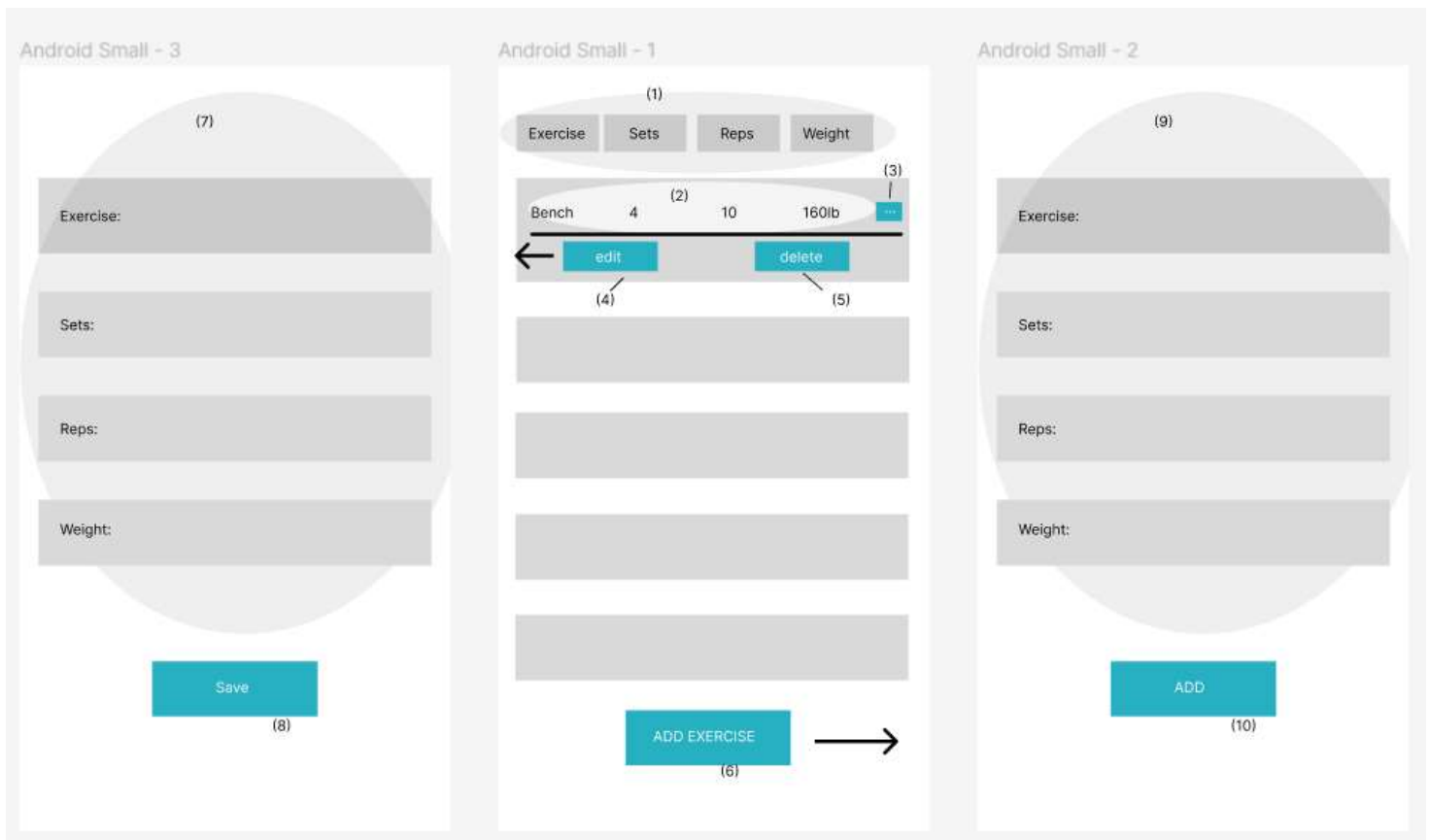
email: (5) email: (12)

6 → EDIT

7 → SAVE

The purpose of this screen is to show the profile of the user from the server. The user is able to click the edit button to the edit page which would save the user input to the server.

- (1) Showing the “name” information of the user.
- (2) Showing the “age” information of the user.
- (3) Showing the “gender” information of the user.
- (4) Showing the “weight” information of the user.
- (5) Showing the “email” information of the user.
- (6) The edit button which takes the user to the edit screen.
- (7) The save button which saves the user input to the server.
- (8) User input of “name”.
- (9) User input of “age”.
- (10) User input of “gender”.
- (11) User input of “weight”.
- (12) User input of “email”.



The purpose of this screen is for users to create and edit their workout plans, so it would be easier for them to stick with the workout plan.

- (1) The column of the workout information including exercise name, sets, reps, weight.
- (2) This is an example of workout information - "bench press".
- (3) The "..." button which would call out options of edit and delete workout.
- (4) The edit button which would bring the user to the edit screen.
- (5) The delete button which would delete this workout by pressing it.
- (6) The add exercise button which would bring the user to the add exercise screen.
- (7) The user can modify the information for editing the exercise.
- (8) The button would save the modified exercise to the server.
- (9) The user can modify the information for creating new exercises.
- (10) The button would add the new exercise to the server.