



# **FLEX**

## **MOBILE APPLICATION**

**Drexel University  
College of Computing and Informatics  
INFO 532 – Software Development  
Course Project Part 3: Software Requirements and Design  
Specification (SRS/SDS)**

**Team Members:**

Danielle Kot . Sesoo Kator Ijir . Shahrar Nizam . Shaima Albugami

**Team Description:**

- **Danielle Kot:** The introduction
- **Sesoo Kator Ijir:** Functional requirements
- **Shahrar Nizam:** External interface requirements, and performance requirements
- **Shaima Albugami:** Overall description, and edited the paper.

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Purpose	1
1.2. Scope	1
1.3. Definitions	1
1.4. References	2
1.5. Overview	2
<b>2. Overall Description</b>	<b>3</b>
2.1. Product Perspective	3
2.2. Product Functions	3
2.3. User Characteristics	3
2.4. Assumptions and Dependencies	3
2.5. Apportioning of requirements	4
<b>3. Specific Requirements</b>	<b>5</b>
3.1. Functional Requirements	5
3.1.1. User	5
3.1.2. Manage Account	5
3.1.3. Manage User Profile	5
3.1.4. Find Potential Matches	6
3.1.5. Match Request	6
3.1.6. Rate matches	6
3.1.7. Reporting	6
3.2. External Interface Requirements	7
3.2.1. User Interface	7
3.3. Performance Requirements	10
3.4. Attributes	11
3.4.1. Availability	11
3.4.2. Interoperability	11
3.4.3. Maintainability	11
3.4.4. Reliability	11
3.4.5. Security	11
3.4.6. Scalability	11
3.4.7. Usability	11

# Chapter 1

## 1. Introduction

### 1.1. Purpose

This document describes the software requirements for a system that matches users who work out with other similar users. It is intended for designers, developers, testers, and maintainers of FLEX.

### 1.2. Scope

The function of the system is to help users who want to work out with a partner match with other users who have similar fitness goals and who practice similar fitness activities.

### 1.3. Definitions

- **Account**

A single account which provides user access to the mobile application. A user can only create an account with a unique email address. An account can only be linked to a single user profile.

- **Fitness activities**

A set of activities that a user can select to indicate what type of activities he/she wants to perform with other users.

- **Fitness goals**

A set of goals that a user can select to indicate his/her motivation for working out.

- **FLEX**

A mobile application that is used by users to search and match with other users to work out with. The mobile application interacts with users to gather information on user profile and user preferences, suggest potential matches for users based on information gathered and allows communication between users once matched.

- **User**

The owner of an account. A user can consist of one or more persons or corporations; the correspondence is not relevant to this problem. The same person who owns a second account using a different email address is considered a different user.

- **User Preference**

Information provided by user to help FLEX mobile application determine potential matches for user. User Preference consist of the user's fitness goals, preferred fitness

activities, preferred gender for potential matches, preferred age range of potential matches, maximum distance of potential matches from user's zip code.

- **User Profile**

Section on mobile application that contains information about a user that is displayed to other users. User Profile contains user's photos, age, gender, distance from the user viewing the profile, user bio, rating (1-5 stars), fitness goals and preferred fitness activities.

- **Match**

An event where a single user accepts another user's request to match. This event allows both users to message one another on the application until either user un-match, deletes account or pause account.

- **Match Request**

A request by a single user to another user to match. This request can be accompanied with a short message by the user who requested the match.

- **Potential Matches**

Any user on the mobile application who another user could request a match with.

- **Un-match**

An event where a user removes an existing match with another user. This removes the ability for both users to message one another.

- **Rating**

An assessment of a user by other users that have matched with the user. The assessment consists of a 1-5 rating. The average of the ratings will be displayed on the user's profile.

- **System**

Covers all components that allow FLEX to function. This includes the FLEX mobile application, the servers and the database.

## **1.4. References**

IEEE Recommended Practice for Software Requirements Specification (IEEE-STD-830-1998). Available at <http://ieeexplore.ieee.org>

## **1.5. Overview**

Section 2 covers the general description of FLEX. Section 3 identifies the specific functional requirements, the non-functional requirements and user interfaces of the FLEX system.

# Chapter 2

## 2. Overall Description

### 2.1. Product Perspective

FLEX will be deployed on IOS for iPhone and Android. Users can download the app on their devices and have access to all features in FLEX after creating their accounts and setting up their profiles. Users can create an account by entering their information and create their profiles by setting up their preferences. Once users set up their profiles, they will gain access to the online platform to find other users who share similar preferences. Matched users can interact by sending private messages to each other via FLEX .

### 2.2. Product Functions

Users can create an account by using their username, email, and password. In the future, users may be allowed to create an account using their Google or Facebook account. Users can log in using their email and password. Users can create their profile by entering their zip code, age, and gender - adding a picture is optional-. Users can set their preferences by specifying the fitness goals, activities, age, and gender for potential matches. FLEX will display a list of potential matches based on user preferences. Users can send a match request to a potential match user. Users can accept or decline the match request. If the message request gets accepted, users can have private conversations.

### 2.3. User Characteristics

- **Users**

Users will have full, non admin access to FLEX once they set up their profiles. They can browse potential matches, send match requests, accept or decline a request, chat with matched users, and rate matched users.

- **Administrator**

The administrator is the only entity that has full admin access to the database and can manage users' account status. The administrator will take action on the account status in case there are many reports on an account or if the account look suspicious. The administrator will ensure that the content of FLEX is clean and up to date.

### 2.4. Assumptions and Dependencies

There are three assumptions on which FLEX is depended on, which are as follows:

- **IOS and Android Operating System:** It is assumed that FLEX runs on iOS and android operating system.
- **Database Management System:** It is assumed that our data will be secured and stored in the ORACLE relational database management system.
- **Google API:** It is assumed that Google maps API is used by FLEX to locate the user location.

## **2.5. Apportioning of requirements**

The following are future features to be implemented in later versions of the system:

- A user can block any user on the platform from accessing their account
- A user can sign up using his/her Google or Facebook account
- A user can share his/her workout schedule with other users.
- A premium account is to be introduced to monetize FLEX.

# Chapter 3

## 3. Specific Requirements

### 3.1. Functional Requirements

The functional requirements are organized according to user activities performed in the system. Requirements are numbered according to the level of priority.

#### 3.1.1. User

All users have the ability to create an account and all information will be securely stored in the database. User information in the database will be used to match users.

#### 3.1.2. Manage Account

- i. Users can create an account with the following required fields:
  - Username
  - Email
  - Password
- ii. Users can delete account permanently
- iii. Users can pause account temporarily which hides their profile from other users
- iv. Users can unpause their paused account
- v. Users can only have one account per email.
- vi. Users can edit account information
- vii. System should provide a login interface with the following required fields:
  - Username
  - Password
- viii. Users can recover password from login interface.

#### 3.1.3. Manage User Profile

- i. Users can create a profile with the following required user information:
  - Zip code
  - Age
  - Gender
  - Photo (Optional)
- ii. Users can enter the following required user preferences in their profile:
  - Fitness Goals
  - Fitness Activity (Optional)
    - Experience level of each Fitness Activity selected
  - Location range for potential matches



- Gender of potential matches
- Age range for potential matches

#### **3.1.4. Find Potential Matches**

- System must be able to display a list of potential matches by user preference
- System must show potential matches location and distance in the display list.
- System must be able to filter potential matches by:
  - Age range
  - Location range
  - Gender selection
- System should allow users sort potential matches by:
  - Best Match
  - Based on similarities with your preference
  - Distance
  - Rating

#### **3.1.5. Match Request**

- System must have a messaging feature
- System matches users by a match request
- Users must get a message notification when another user sends a message request
- Users can accept match request
- Users can deny match request
- Matches communicate through messaging feature
- Users can unmatched

#### **3.1.6. Rate matches**

- System must have a 5 point rating system
- Users can rate matches only if they match

#### **3.1.7. Reporting**

Users can report other users

## 3.2. External Interface Requirements

### 3.2.1. User Interface

The user interface must be simple and allow the user to navigate inside the application without the need for a manual. Elements of the interface must sit well together on a smartphone screen and maintain a color theme. Users should be able to tap, slide and type using the smartphone keyboard into the application. The following screenshots are examples of what the different screens could look like as the user interacts with the application.

Fig 3a. Sign Up

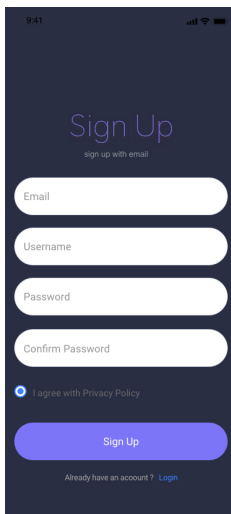


Fig 3b. Create Profile

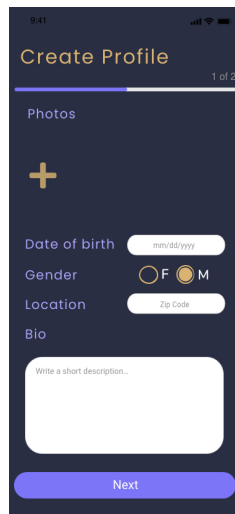


Fig 3c. Create Profile

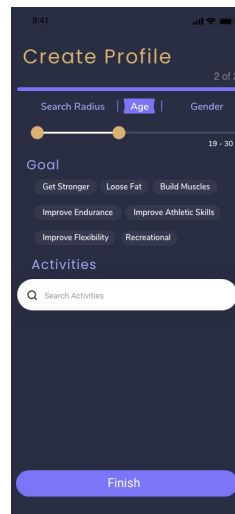


Fig 3d. Welcome Page



Fig 3e. Login Page

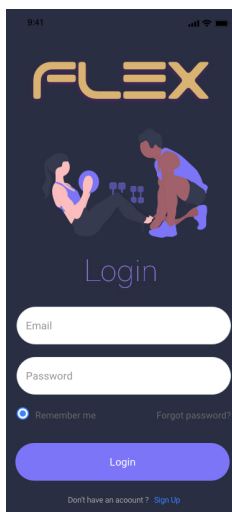


Fig 3f. Search Page

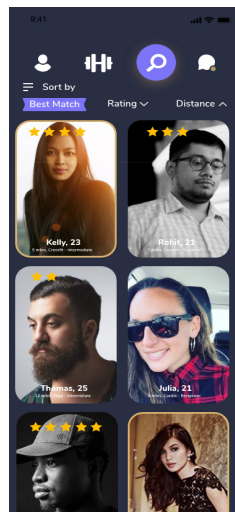


Fig 3g. View Profile

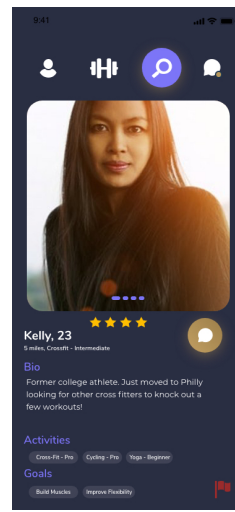


Fig 3h. Send Request

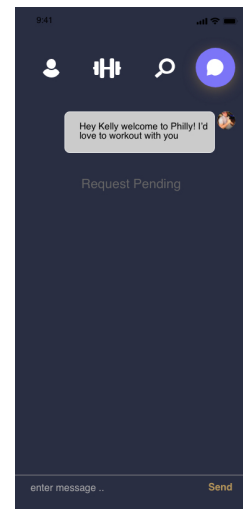


Fig 3i. Inbox View

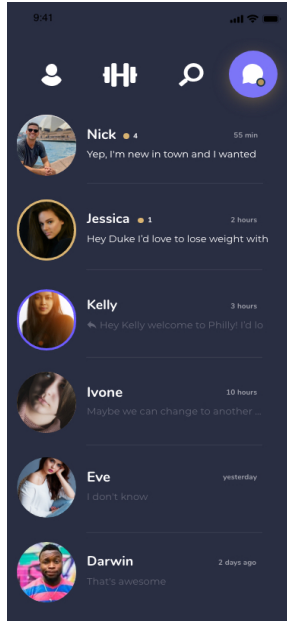


Fig 3j. Receive Request

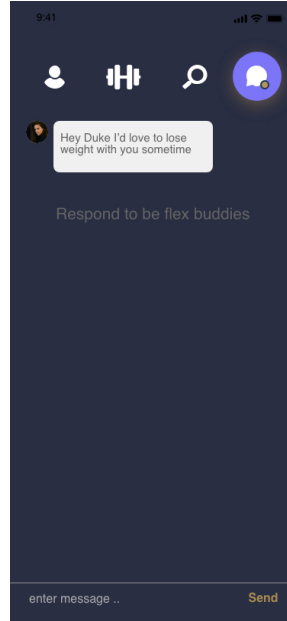


Fig 3k. Chatbox

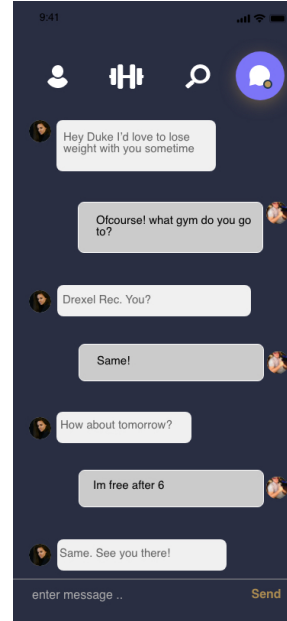


Fig 3l. Mached User

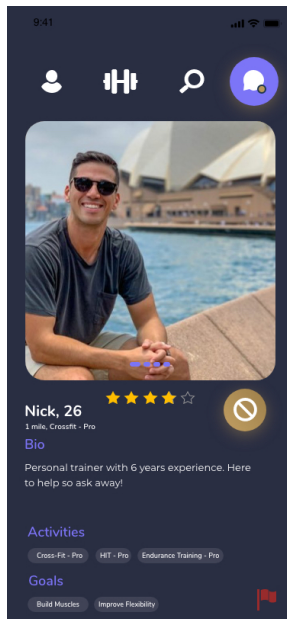


Fig 3m. Report User

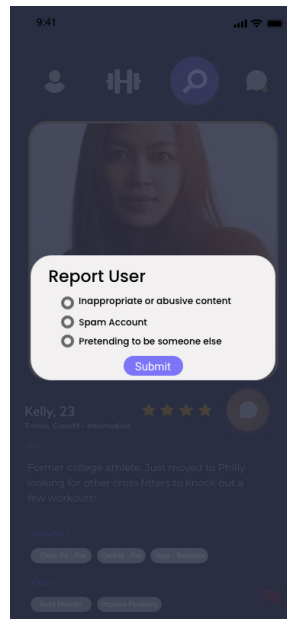


Fig 3n. Un-match User

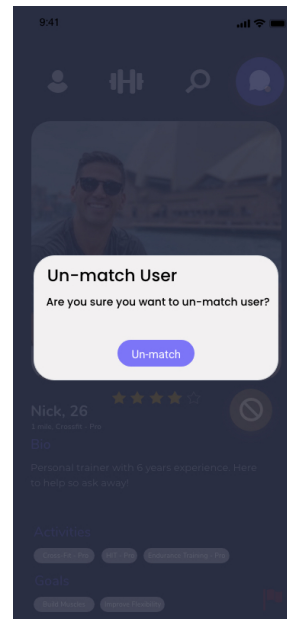


Fig 3o. Profile and Account Settings

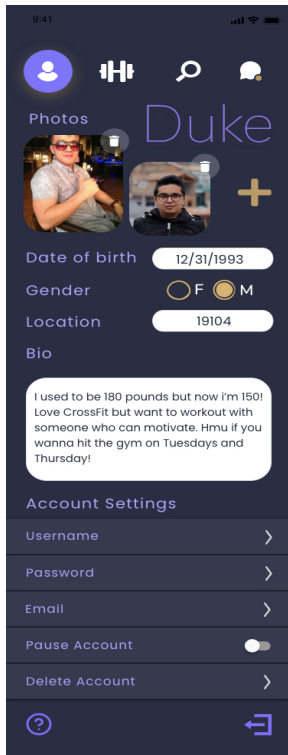


Fig 3p. Change Username

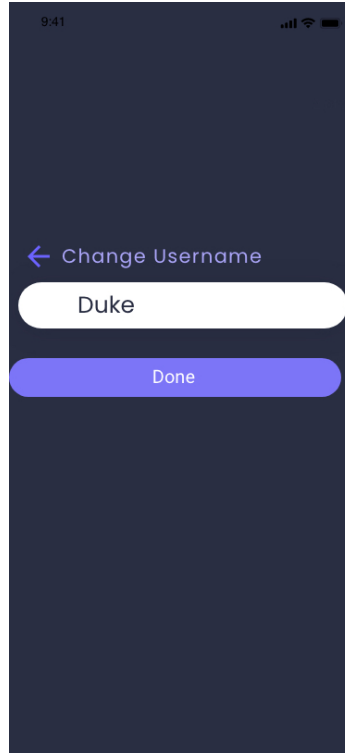


Fig 3q. Change Password

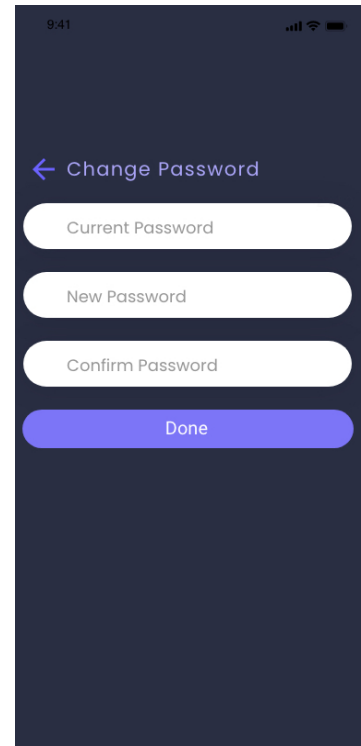


Fig 3r. Change Email

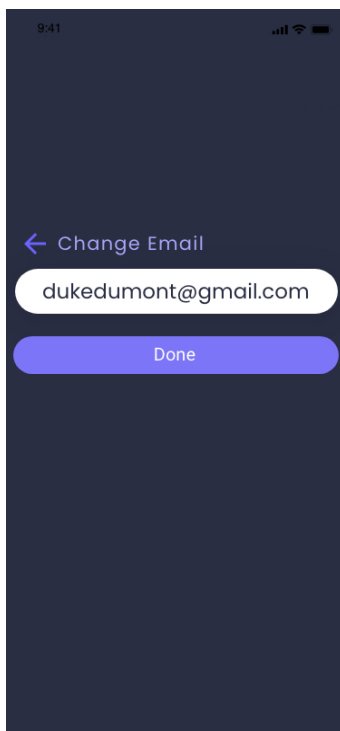


Fig 3s. Delete Account

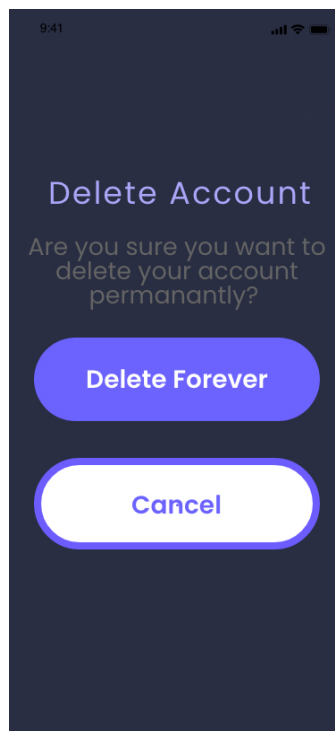


Fig 3t. Preferences Page

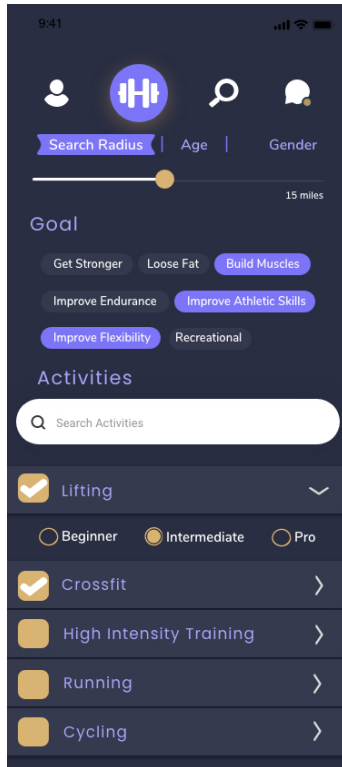
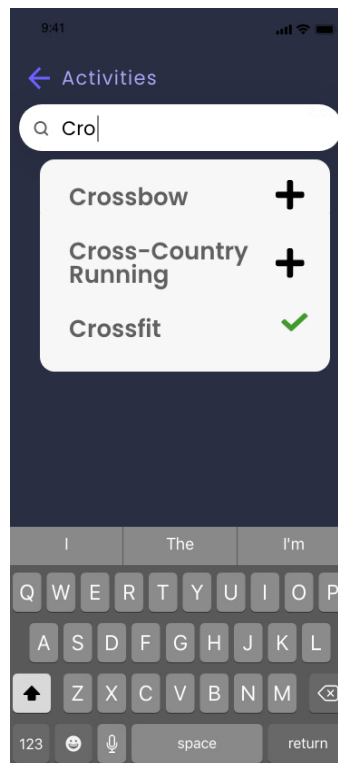


Fig 3u. Add Activities



### 3.3. Performance Requirements

#### Performance Requirement 1

- *Description*

FLEX must launch on a smartphone device in less than 2 seconds.

#### Performance Requirement 2

- *Description*

*All requests within the app must respond in less than 1 second.*

#### Performance Requirement 3

- *Description*

At login, if the user is not authenticated in 1 minute, cancel login request.

#### Performance Requirement 4

- *Description*

*Crash rate must be less than 2% at all times.*

#### Performance Requirement 5

■ *Description*

*FLEX must refresh when user is inactive for 2 minutes.*

### **3.4. Attributes**

#### **3.4.1. Availability**

- i. The FLEX server must be available 365 days a year with at least 99% availability.
- ii. User must be able to access all FLEX functionalities from anywhere in the United States.

#### **3.4.3. Interoperability**

- i. FLEX must be compatible on both Android and iOS devices.
- ii. FLEX must be functional when operating systems are updated.

#### **3.4.3. Maintainability**

- i. FLEX must be able to update without removing, changing or deleting user data.
- ii. There must be an admin dashboard interface for administrators to support the system.

#### **3.4.3. Reliability**

- i. Data displayed within FLEX must be consistent for all users.
- ii. There must be a backup system to support downtime and maintenance of the system.

#### **3.4.3. Security**

- i. The system must identify and correctly authenticate users at login.
- ii. Stored and transferring messages between the users must be encrypted.
- iii. System must be able to remove all user data when account is deleted.

#### **3.4.4. Scalability**

- i. System must be able to handle workload when all users are using FLEX at the same time without impacting performance.
- ii. Performance must remain constant with more users and functionalities.

#### **3.4.3. Usability**

- i. Users must be able to perform all use cases without a user manual.