Team 26

9/29/2017

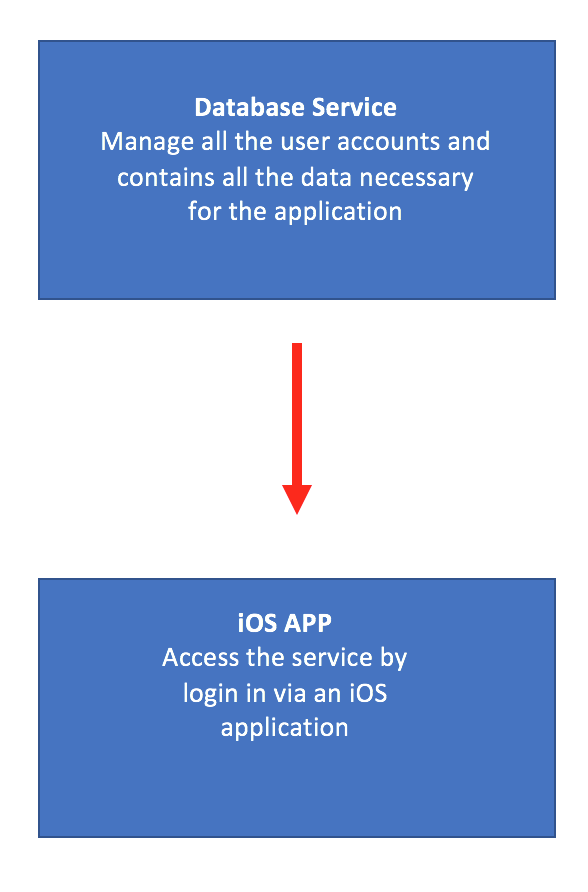
**Incremental and Regression testing**

Team GymWorkout

Harsha Lingareddy, Manoj Polisetti, Dhiman Swadia, Sanat Mouli, Vishaal Bommena

**Incremental and Regression Testing**

1. Classification of Components
   1. Define all components
      1. Overall



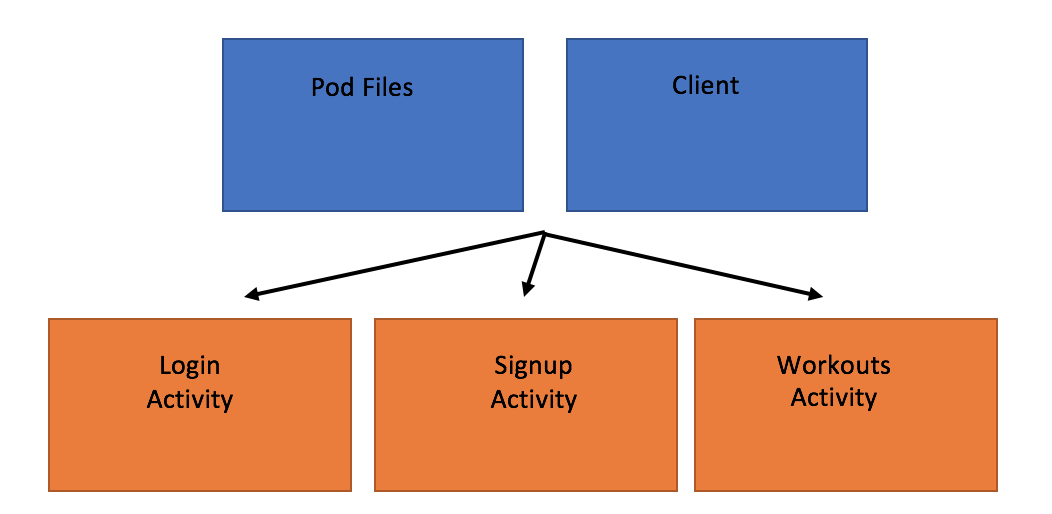
The overall system consists of two major components. A Database service which is responsible for managing all of the user accounts as well as performing the main functionality of the system calls. The iOS application allows the users to access the functionality of the service by logging in through the mobile app.

Database Service

* Input: Actions from the iPhone application
* Output: Data to display in the iPhone application
* Dependencies: Next section provide more information regarding this
* Dependents: iPhone application

iPhone Application

* Input: User actions
* Output: Displaying data based on the command from the user
* Dependencies: iOS Storyboards
* Dependents: Users
  + 1. iOS Application



The iOS app consist of many activities. Each activity allows the users to access the different parts of the system. Which includes the following: login process, the signing up a new user and also accessing the curated workout list. The activities make use of pod files for packages and libraries as well as a client class to handle communication with database.

Activities (All in Orange)

* Input: User actions via the UI interface
* Output: Updating the interface based off of commands
* Dependencies: Pod Files and Client
* Dependents: Users

Pod Files

* Input: Construction of class
* Output: Accessing properties in the controller files
* Dependencies: None
* Dependents: Controllers

Client

* Input: Method calls to FireBase database
* Output: Response message for activities resulting from connection to database
* Dependencies: Podfile
* Dependents: Controllers
  1. Which form of incremental testing did you follow?
     1. While we attempted to use multiple forms of incremental testing, we concentrated on primarily implementing top-down testing. This helped us ensure that the individual components of the application (such of communication between the application and the database service) was functioning as required, while allowing us to work substitute the unavailable components (functionality to create personal workouts) for stubs so that they can be completed at a later point of time. This form of incremental testing also allowed us to postponing focus on smaller details at a later point of time.

Incremental and Regression Testing

|  |
| --- |
| Incremental Testing |

|  |
| --- |
| iOS Application |

|  |  |  |  |
| --- | --- | --- | --- |
| Defect | Description | Severity | How to Correct |
| 1 | Log in does not respond if user is not able to communicate with the database | 3 | Add timeout to client and print an error in dialog box for user to see that timeout has occurred |
| 2 | Log in crashed if given empty username or password to send to web service | 2 | When checking username and password check for empty length before attempting other operations |
| 3 | Check if email format is valid | 3 | We added check to see if email is int the valid format |

|  |
| --- |
| Database Service |

|  |  |  |  |
| --- | --- | --- | --- |
| Defect | Description | Severity | How to Correct |
| 1 | IOS application client does not get a response from server | 2 | Must ensure that client receives a message regardless of the server status. |
| 2. | Client does not timeout when connection takes too long | 3 | Must ensure that the client receives a timeout message in the case of not connecting to server |
| 3. | Client does not receive error notification from the server in the case of login credential mismatch. | 1 | Must ensure that username and password and other signup details are validated before posting to database. |

|  |
| --- |
| Regression Testing |

|  |
| --- |
| iOS Application |

|  |  |  |  |
| --- | --- | --- | --- |
| Defect | Description | Severity | How to Correct |
| 1 | Navigation Controller is not routing to the correct location | 1 | Use right identifier for the segues |
| 2 | Password length if too long app crashes | 2 | Their was a limit set at 32 for the password limit. |
| 3 | User authentication doesn't persist even if the user is logged in and has a valid session token | 3 | Added check to the database for valid authentication credentials. |

|  |
| --- |
| Database Service |

|  |  |  |  |
| --- | --- | --- | --- |
| Defect | Description | Severity | How to Correct |
| 1. | App crashes if the data fetch fails | 1 | Change the sequence of the fetch so that the data is fetched before the views are loaded. |
| 2. | Malformed database due to bad request | 2 | Added different nodes to the JSON tree parser to differentiate the objects. |
| 3. | Database returned a String instead of an Object | 3 | Change firebase schema to handle an object. |

Update Product Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Requirement | Status | Sprint |
| 1 | User can log in to the application | Complete | 1 |
| 2 | User can sign in to the application | Complete | 1 |
| 3 | User can sign out of the application | Complete | 1 |
| 4 | User can choose a preset workout based on muscle groups from a list of preset workouts | Complete | 1 |
| 5 | User can switch between tabs using a tab bar | Complete | 1 |
| 6 | User can search for selected workouts by name or muscle group | Incomplete | In-Progress: Moved to Sprint 2 |
| 7 | User can view a details about individual workouts | Incomplete | In-progress: moved to Sprint 2 |
| 8 | User can choose to create a workout and edit the number of sets and reps for exercises | Incomplete | In-progress: moved to Sprint 2 |
| 9 | User can view a list of created workouts | Incomplete | In-progress: moved to Sprint 2 |
| 10 | User can delete workouts from the workout list | Incomplete | Moved to Sprint 2 |
| 11 | User can edit a profile picture | Incomplete | Planned for Sprint 2 |
| 12 | User can choose to look for workout partners | Incomplete | Planned for Sprint 2 |
| 13 | User can choose to look for workout partners on a map view | Incomplete | Planned for Sprint 2 |
| 14 | User can message workout partners | Incomplete | Planned for Sprint 2 |
| 15 | User can check for the Corec timings | Incomplete | Planned for Sprint 2 |
| 16 | User receives daily notifications to go to the gym | Incomplete | Planned for Sprint 2 |
| 17 | User can log in using touch id | Incomplete | Planned for Sprint 2 |