

# **Stay Fit (A Fitness iPhone App)**

**by**

**Prashant Gohel (ID No. 11CEUOD005)**

**&**

**Sarju Hansaliya (ID No. 102043)**

**A project submitted**

**in**

**partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER ENGINEERING**

**Internal Guide**

*Prof. Nainesh Patel*

*Associate Professor*

*Dept. of Comp. Engg. DDU*

**External Guide**

*Mr. Kuldeepsinh Kumpavat*

*Senior Software Engineer*

*Infostretch Solutions Private Limited*



**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University  
April 2014**

# **CERTIFICATE**

This is to certify that the project work titled

## **Stay Fit (A Fitness iPhone App)**

is the bonafide work of

**Prashant Gohel (ID No.11CEUOD005) and Sarju Hansaliya (ID No. 102043)**

carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session December 2013 to April 2014.

Prof. Nainesh Patel  
Associate Professor,  
Dept. of Computer Engg.  
Engg.

Prof. C.K.Bhensdadia  
Head of Dept.,  
Dept. of Computer



**Faculty of Technology  
Department of Computer Engineering  
Dharmsinh Desai University**

## **Stay Fit (A Fitness iPhone App)**

Project by

**Prashant Gohel (ID No. 11CEUOD005) and Sarju Hansaliya (ID No. 102043)**

## **ABSTRACT**

‘Stay Fit’ is basically an iPhone App which covers the new emerging technology of market named BLE (Bluetooth Low Energy).

BLE is an application-friendly technology supported by every major operating system. The technology costs less and offers flexible development architecture for creating applications to bring everyday objects like heart-rate monitors, toothbrushes, and shoes into the connected world and have them communicate with applications that reside on the Bluetooth Smart compatible smartphones, tablets, or similar devices those consumers already own. This means Bluetooth Smart developers are limited only by their imagination.

‘Stay Fit’ uses a peripheral device, which has implemented the HRM (Heart Rate Monitor) profile of Bluetooth smart protocol. The application gets user’s heart rate in bps unit from the device and also records the location details and allows users to track his/her fitness, all in one app. In addition to that, the application keep record of user’s daily activity and also shows a statistics of all workouts that user has undergone to. The application also uses GPS to accurately measure the distance covered and speed of user during workout. Over all, ‘Stay Fit’ helps user to stay fit.

## **ACKNOWLEDGEMENT**

It gives us immense pleasure and satisfaction in presenting this report of System development Project undertaken during the 8<sup>th</sup> semester of B.Tech.

As it is the first step into our Professional Life, we would like to take this opportunity to express our sincere thanks to several people, without whose help and encouragement, it would be unfeasible for us to carry out the desired work.

We would like to thank to **Mr. Shailesh Kanzariya** (*Engineering Manager*) for giving us an opportunity to work with one of the esteemed organization of the world. An enviable work culture and an environment that encourages creativity and innovation have inculcated in us a sense of discipline and perseverance.

We are indebted to **Mr. Kuldeepsinh Kumpavat** (*External Guide*), **Mr. Chintan Prajapati** (*Supporting Guide*) for guiding us throughout the project development and helped us conceptualize the idea of the project and helped us streamline our work by providing us with facility required.

From the bottom of our heart, I would like to express my sincere thanks to our Head of Department **Prof. C. K. Bhensdadia** and our internal guide **Prof. Nainesh Patel**, who gave us an opportunity to undertake such a great challenging and innovative work. We are grateful to them for their guidance, encouragement, understanding and insightful support in the development process.

Finally, we would like to thank to all Infostretch Private Limited (Ahmedabad) employees, all faculty members of our college, our friends and our family members for providing their support and continuous encouragement throughout the project.

With sincere regards,

Prashant Gohel  
& Sarju Hansaliya

## Table of Contents

Title	Page
<b>Certificate</b> .....	i
<b>Abstract</b> .....	ii
<b>Acknowledgments</b> .....	iii
<b>List of Figures</b> .....	x
<b>List of Tables</b> .....	xiv
<b>Company Profile.</b> .....	xvi
<b>Chapter 1: Introduction</b> .....	1
1.1 Project Definitions.....	2
1.2 Project Overview.....	2
1.3 Project Objective.....	3
1.4 Scope.....	3
1.5 Organization.....	3
1.6 Platform.....	4
1.7 Technology Used.....	4
1.8 Database.....	4
1.9 Methodology.....	4
1.10 Tools used.....	5
1.11 Project Execution Type.....	5
1.12 Project Type.....	5
1.13 Project Duration.....	5

<b>Chapter 2: About the System.....</b>	<b>6</b>
2.1 About ‘Stay Fit’ .....	7
2.2 Requirement for ‘Stay Fit’.....	7
2.2.1 Hardware Requirements.....	7
2.2.2 Software Requirements.....	7
2.3 Feasibility Study.....	8
2.3.1 Technical Feasibility.....	8
2.3.2 Time Schedule Feasibility.....	8
2.3.3 Operational Feasibility.....	8
2.3.4 Implementation Feasibility.....	8
2.4 Project Planning.....	9
2.4.1 Project Development Approach & Justification.....	9
2.4.2 Milestones & Deliverables.....	11
2.4.3 Roles & Responsibilities.....	11
2.4.4 Group Dependencies.....	13
2.4.5 Project Scheduling Chart.....	14
<b>Chapter 3: System Analysis.....</b>	<b>15</b>
3.1 Introduction.....	16
3.1.1 Purpose.....	16
3.1.2 Scope.....	16
3.1.3 How to Use This Document.....	16

3.1.4 List of Abbreviations.....	16
3.2 Requirement Specifications.....	17
3.2.1 Functional Requirements.....	17
3.2.2 Non Functional Requirements.....	23
3.3 Use Case Diagram.....	25
3.4 Class Diagram.....	26
3.5 Sequence Diagrams.....	27
3.6 State Diagrams.....	31
3.7 Activity Diagram.....	34
 <b>Chapter 4: Design.....</b>	 35
4.1 Entity Relationship Diagram.....	36
4.2 Data Dictionary.....	37
4.3 Front-End Interface, Validations & Navigation Design.....	39
 <b>Chapter 5: Implementation.....</b>	 42
5.1 Implementation Environment.....	43
5.2 Technology & Literature Review.....	44
5.2.1 Xcode 5.....	44
5.2.2 Ios.....	45
5.3 Modules Specification.....	46
5.3.1 User Profile.....	46

5.3.2 Workout.....	46
5.3.3 Reminder.....	47
5.3.4 BLE Connection.....	47
5.3.5 HR Distributor.....	47
5.3.6 Voice Assistance.....	47
5.3.7 Music Player.....	47
5.3.8 Social Network Sharing.....	48
5.4 Coding Standards.....	48
5.4.1 General Principals.....	48
5.4.2 Methods.....	49
5.4.3 Properties.....	50
 <b>Chapter 6: Testing.....</b>	 52
6.1 Testing Plan.....	53
6.2 Testing Strategies.....	54
6.2.1 Unit Testing.....	54
6.3 Testing Methods.....	55
6.4 Test Cases.....	56
6.4.1 Save User Profile .....	56
6.4.2 Edit User Profile.....	56
6.4.3 Connect BLE Peripheral.....	57
6.4.4 Disconnect BLE Peripheral.....	57

6.4.5 Save Workout Goal.....	58
6.4.6 Enable Workout Goal.....	58
6.4.7 Disable Workout Goal.....	59
6.4.8 Play Music.....	59
6.4.9 Delete Today's Workout Report.....	60
6.4.10 Delete Random Workout Report.....	60
6.4.11 Share Workout Report.....	61
6.4.12 Share Workout Statistics.....	61
6.4.13 Plot HR.....	62
6.4.14 Add New Reminder.....	62
6.4.15 Edit Reminder.....	63
6.4.16 Delete Reminder.....	63
<b>Chapter 7: User Manual.....</b>	<b>64</b>
7.1 Splash Screen.....	65
7.2 Registration.....	66
7.3 Connection Manager.....	70
7.4 Location Services.....	73
7.5 Workout.....	75
7.5.1 Workout Path.....	75
7.5.2 Workout Reports.....	77
7.6 Workout Goal.....	82

7.7 Workout Reminder.....	87
7.7.1 Reminder Permission.....	87
7.7.2 Add Reminder.....	88
7.7.3 Edit/Delete Reminder.....	95
7.8 Heart Rate Monitor.....	100
7.9 Music Player.....	103
7.10 Reset All Data.....	105
7.11 Social Network Sharing.....	107
<b>Chapter 8: Conclusion and Future Extensions.....</b>	<b>110</b>
8.1 Limitations.....	111
8.2 Discussion.....	111
8.2.1 Self Analysis of Project Viabilities.....	111
8.2.2 Problems Encountered & Possible Solutions.....	112
8.2.3 Summary of Work Done.....	113
8.3 Conclusion.....	113
8.4 Future Extensions.....	114
<b>Bibliography.....</b>	<b>115</b>
<b>Project Guide Detail.....</b>	<b>116</b>

## List of Figures

Figure: 2.1	Iterative Waterfall Model.....	9
Figure: 3.1	Usecase Diagram.....	25
Figure: 3.2	Class Diagram.....	26
Figure: 3.3	Sequence Diagram for Start Workout.....	27
Figure: 3.4	Sequence Diagram for Connect BLE Device.....	28
Figure: 3.5	Sequence Diagram for Save Workout Goal.....	29
Figure: 3.6	Sequence Diagram for Add New Reminder.....	30
Figure: 3.7	State Chart Diagram for Peripheral Object.....	31
Figure: 3.8	State Chart Diagram for Reminder Object.....	32
Figure: 3.9	State Chart Diagram for Workout Object.....	33
Figure: 3.10	Activity Diagram for Start Workout.....	34
Figure: 4.1	Entity Relationship Diagram.....	36
Figure: 4.2	Dashboard Wireframe.....	39
Figure: 4.3	User Profile Wireframe.....	39
Figure: 4.4	Workout Goal Wireframe.....	40
Figure: 4.5	Connection Manager Wireframe.....	40
Figure: 4.6	HR Monitor Wireframe.....	41
Figure: 5.1	Xcode 5 Screenshot.....	44
Figure: 5.2	IOS Structure.....	45
Figure: 6.1	Testing plan.....	53
Figure: 7.1.1	App Icon Screenshot.....	65
Figure: 7.1.2	Launch Screen Screenshot.....	66
Figure: 7.2.1	First Screen Screenshot.....	66
Figure: 7.2.2	Sample Filled Profile Screenshot.....	67
Figure: 7.2.3	Incomplete Profile Screenshot.....	67
Figure: 7.2.4	Validations Screenshot.....	68

Figure: 7.2.5	DOB Picker Screenshot.....	68
Figure: 7.2.6	Menu Button Screenshot.....	69
Figure: 7.2.7	Menu Screenshot .....	69
Figure: 7.3.1	Navigate To Connection Manager Screenshot.....	70
Figure: 7.3.2	Connection Manager Screenshot.....	70
Figure: 7.3.3	Scanning Screenshot.....	71
Figure: 7.3.4	Connect to Peripheral Screenshot.....	71
Figure: 7.3.5	Disconnect Peripheral Screenshot.....	72
Figure: 7.3.6	Successful Connection Screenshot.....	72
Figure: 7.4.1	Location Services Access grant Screenshot.....	73
Figure: 7.4.2	Turn On Location Services Screenshot.....	73
Figure: 7.4.3	Access Permission for Stay Fit Screenshot.....	74
Figure: 7.5.1.1	Dashboard Screen Screenshot.....	75
Figure: 7.5.1.2	Workout Path Screenshot.....	75
Figure: 7.5.1.3	Map Type Screenshot.....	76
Figure: 7.5.1.4	Hybrid Map Screenshot.....	76
Figure: 7.5.1.5	Menu During Workout Screenshot.....	77
Figure: 7.5.2.1	Navigation to Today's Report Screenshot.....	77
Figure: 7.5.2.2	Today's Workout Reports Screenshot.....	78
Figure: 7.5.2.3	Workout Details Screenshot.....	78
Figure: 7.5.2.4	Delete Report Screenshot.....	79
Figure: 7.5.2.5	Workout Report Navigation Screenshot.....	79
Figure: 7.5.2.6	Workout Reports Screenshot.....	80
Figure: 7.5.2.7	Pick a Date for Workout Reports Screenshot.....	80
Figure: 7.5.2.8	Navigation to Statistics Screenshot.....	81
Figure: 7.5.2.9	Statistics Screenshot.....	81
Figure: 7.6.1	Navigation to Workout Goal Screenshot.....	82
Figure: 7.6.2	Workout Goal Screenshot.....	82
Figure: 7.6.3	Workout Goal Validation Screenshot.....	83
Figure: 7.6.4	Setting Sample Workout Goal Screenshot.....	83
Figure: 7.6.5	Workout Goal Saved Screenshot.....	84
Figure: 7.6.6	Enable Workout Goal Screenshot.....	84
Figure: 7.6.7	Workout Goal Completion Notification 1 Screenshot.....	85

Figure: 7.6.8	Workout Goal Completion Notification 2 Screenshot.....	85
Figure: 7.6.9	Workout Goal Completion Notification 3 Screenshot.....	86
Figure: 7.6.10	Workout Goal Setting Validation Screenshot.....	86
Figure: 7.7.1.1	Reminder Settings Screenshot.....	87
Figure: 7.7.1.2	Reminder Access Screenshot.....	87
Figure: 7.7.2.1	Navigation to Workout Reminders Screenshot.....	88
Figure: 7.7.2.2	Workout Reminder List Screenshot.....	88
Figure: 7.7.2.3	Add New Reminder Screenshot.....	89
Figure: 7.7.2.4	Reminder Repeat Pattern Screenshot.....	89
Figure: 7.7.2.5	Reminder Alarm Pattern Screenshot.....	90
Figure: 7.7.2.6	Reminder Save Confirmation Screenshot.....	90
Figure: 7.7.2.7	List of Reminders Screenshot.....	91
Figure: 7.7.2.8	Reminder Alarm Notification Screenshot.....	91
Figure: 7.7.2.9	Fitness Calendar Screenshot.....	92
Figure: 7.7.2.10	Reminder in Native App Screenshot.....	92
Figure: 7.7.2.11	Reminder Details Screenshot.....	93
Figure: 7.7.2.12	Repeat Pattern in Native App Screenshot.....	93
Figure: 7.7.2.13	Add new Reminder in Native App Screenshot.....	94
Figure: 7.7.2.14	Reminder Fetch Screenshot.....	94
Figure: 7.7.3.1	Select Reminder Screenshot.....	95
Figure: 7.7.3.2	Edit Reminder Screen Screenshot.....	95
Figure: 7.7.3.3	Change Repeat Pattern Screenshot.....	96
Figure: 7.7.3.4	Save Changes Confirmation Screenshot.....	96
Figure: 7.7.3.5	Changes Updated Screenshot.....	97
Figure: 7.7.3.6	Enable Disable Reminder Screenshot.....	97
Figure: 7.7.3.7	Repeat Pattern Missed Out Screenshot.....	98
Figure: 7.7.3.8	Delete Reminder Confirmation Screenshot.....	98
Figure: 7.7.3.9	Reminder Deleted Screenshot.....	99
Figure: 7.7.3.10	Reminder Deleted in Native App Screenshot.....	99
Figure: 7.8.1	Navigation to Heart Rate Monitor Screenshot.....	100
Figure: 7.8.2	Filling Filtering Details Screenshot.....	100
Figure: 7.8.3	HR Graph (hh:mm) Screenshot.....	101
Figure: 7.8.4	HR Graph (MMM-yy) Screenshot.....	101

Figure: 7.8.5	HR graph (hh:mm:ss) Screenshot.....	102
Figure: 7.9.1	Showing Music Player Controls Screenshot.....	103
Figure: 7.9.2	Add Songs to Playlist Screenshot.....	103
Figure: 7.9.3	Music Being Played Screenshot.....	104
Figure: 7.9.4	Playlist Screenshot.....	104
Figure: 7.10.1	Reset All Data Screenshot.....	105
Figure: 7.10.2	Confirmation of Resetting Data Screenshot.....	105
Figure: 7.10.3	Progress of Resetting Data Screenshot.....	106
Figure: 7.10.4	Progress of Resetting Data Completed Screenshot.....	106
Figure: 7.11.1	Sharing Option Screenshot.....	107
Figure: 7.11.2	Account Configuration Screenshot.....	107
Figure: 7.11.3	Facebook Dialog (Statistics) Screenshot.....	108
Figure: 7.11.4	Facebook Dialog (Workout) Screenshot.....	108
Figure: 7.11.5	Account Configuration Missing Screenshot.....	109

## **List of Tables**

Table: 2.1	Milestones & Deliverables.....	11
Table: 2.2	Project Schedule Chart.....	14
Table: 3.1	List of Abbreviations.....	16
Table: 4.1	User_Details Schema.....	37
Table: 4.2	HR_Details Schema.....	37
Table: 4.3	Location_Details Schema.....	37
Table: 4.4	WO_Goal_Details Schema.....	37
Table: 4.5	WO_Details Schema.....	38
Table: 5.1	Comparison between GUI & NON-GUI.....	43
Table: 5.2	Comparison between Single User & Multi User.....	44
Table: 6.1	Save User Profile Test Case.....	56
Table: 6.2	Edit User Profile Test Case.....	56
Table: 6.3	Connect BLE Peripheral Test Case.....	57
Table: 6.4	Disconnect BLE Peripheral Test Case.....	57
Table: 6.5	Save Workout Goal Test Case.....	58
Table: 6.6	Enable Workout Goal Test Case.....	58
Table: 6.7	Disable Workout Goal Test Case.....	59

Table: 6.8	Play Music Test Case.....	59
Table: 6.9	Delete Today's Workout Report Test Case.....	60
Table: 6.10	Delete Random Workout Report Test Case.....	60
Table: 6.11	Share Workout Report Test Case.....	61
Table: 6.12	Share Workout Statistics Test Case.....	61
Table: 6.13	Plot HR Test Case.....	62
Table: 6.14	Add New Reminder Test Case.....	62
Table: 6.15	Edit Reminder Test Case.....	63
Table: 6.16	Delete Reminder Test Case.....	63

# **COMPANY PROFILE**

## **Infostretch Solutions Private Limited**



### **About**

Infostretch is a leading provider of mobile and enterprise QA services and solutions. Our offerings range from enterprise QA, mobile application development, testing, and automation to certification and sustenance. We've been providing expert solutions and services to various large and strategic players in the mobile ecosystem including enterprises, carriers, app stores, VAS, and ISV's to actualize their mobile and QA strategy very successfully.

We offer the most cost effective, secure, and scalable models to service our clients across various verticals such as banking, healthcare, education, and entertainment to name a few. Our innovations have allowed us to provide an SLA of 24hours and the ability to scale and test over 2,000 apps a day.

Infostretch was formed in 2004 by a team of highly qualified and experienced professionals known in the software QA world. The team came together with the goal of leveraging their international industry experience to build a global, comprehensive services organization with the highest quality of services.

### **Vision**

Providing cost effective QA and Test Automation services to global customer base and have managed to grow significantly since its inception and create a distinct brand for its services.

Provide value-added solutions that significantly improve the quality of software by leveraging technology

### **Mission**

Serve customer needs globally by innovating market-leading information technology products and solutions to deliver the highest quality in software while leveraging business best practices and integrated world-class talent.

STRETCH the reach of technology to deliver superior solutions, STRETCH the boundaries of geographies to deliver cost-effective solutions, STRETCH our customer's efforts with expertise and sincerity to deliver meaningful Information to customers so they can make Informed decisions. Doing more with less while stretching as much as possible.

## **Values:**

### Customer Satisfaction

We put forth the extra effort to ensure our customers success – even under the most challenging conditions. Customer Satisfaction is our top priority in all of our engagements and we track employee performance metrics that are centered on delivering customer satisfaction.

### Commitment to Employees

We nurture teamwork, candidness and wisdom among our employees to get the best out of each one of them. Management is devoted and committed to growth, satisfaction and performance. We strongly believe that happy employees produce happy customers.

### Passion for What We Do

We are passionate about what we do. We make sure every employee is happy with their work and feels pride in accomplishing the task at hand. We set high standards for everyone in the company and ensure that we attain them.

### Thinking Outside the Box

Technology, Processes, Decisions: Our innovative thinking is rooted in all aspects of our existence. We strive to foster innovation, top down and bottom up throughout the organization.

### Increasing Value for our Share Holders

While keeping Customer satisfaction and employee commitment as top priorities, we put genuine effort into creating valuable relationships, offering reusable IP and marketing brand strategy, resulting in greater value for our shareholders.

## Quality Strategy Process

Discover – Define – Demonstrate – Realize



### Some of the many accolades earned by Infostretch include:

<b>2008</b>	1. Gartner Top 20 QA Service Provider Exceptional QA Services and Solutions for Enterprise, Mobile and SaaS.
<b>2009</b>	2. Business Journal: Awarded "Fastest 30 Privately Owned Companies in Bay Area".
<b>2010</b>	3. Business Journal: Awarded "Fastest 30 Privately Owned Companies in Bay Area". 4. Lead411: Ranked among "TOP 200 Fastest Growing Companies" in USA in 2010. 5. GESIA: Rising Star in Best Software Product & Service Company. 6. Awarded Inc 500/5000 Company of the Year 2010. 7. Best of Santa Clara Award for Mobile & Software Testing.
<b>2011</b>	8. GESIA: Silver Award for best service company. 9. The company is among the 20 fastest growing technology companies in Silicon Valley, California. 10. Forbes: Among Top Most Promising Companies in US.
<b>2012</b>	11. GSIA: Platinum Award for Best Mobile Application Development Company. 12. NASSCOM Applauds Infostretch's Growth.
<b>2013</b>	13. Named an Ernst & Young Entrepreneur of the Year® 2013 Semifinalist Northern California. 14. Named a Fast 50 Asian American Business for 2013. 15. Ranked 44th in the Prestigious Tech 200 List

## **Chapter 1**

### **Introduction**

---

## 1.1 Project Definition

An iOS application that will allow user to track his/her workout process and help them to stay fit. It also will be able to connect with any BLE peripheral which implements HRM profile. It record user's heart rate and tracks user's location during workout so that user can improve his/her workout process.

This application will also remind the user time to time to undergo workout and also allow user to set the reminder on desired time. The application will also provide the facility of music so that user can relax and also provide voice assistance as feedback during workout.

## 1.2 Project Overview

Obesity is the most prevalent, fatal, chronic, relapsing disorder of the 21st century. Obesity is a leading cause of world's mortality, morbidity, disability, healthcare utilization and healthcare costs. It is likely that the increase in obesity will strain our healthcare system with millions of additional cases of diabetes, heart disease and disability. Significantly, excess adiposity or obesity causes insulin secretion, which can cause insulin resistance that leads to type-2 diabetes.

Regular physical activity will help you to avoid these things. Physical activity is essential to prevent and reduce risks of many diseases and improve physical and mental health. It can even help you live longer—research from the American Journal of Preventative Medicine indicates that regular exercise can add up to five years to your life.

Since regularity in physical activity is very important, the development of the fitness application that can run in your mobile device is necessary.

This mobile application can be used to track different parameters during workout. This app will be able to connect to BLE enabled Heart Rate Monitor. This monitor will continuously send user's Heart Rate data to application. This application will keep track of Heart Rate data along with user's Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map. User will be able to set his/her workout goals through this application. Application will notify user on completion of any goal. User will be able to view/share his/her daily/weekly/monthly workout statistic from app. This app will remind user for Workout if he/she has set any reminder. User will be able to Play music during workout.

### **1.3 Project Objective**

‘Stay Fit’ is charged with the responsibility of being helpful in being fit by including following activities.

- Set Workout Goals
- Set Workout Reminder
- HR Monitor Connection
- Continuous Hear Rate Monitoring
- Undergo Workout
- Workout Report
- Music Player
- Voice Assistance
- Social Sharing

### **1.4 Scope**

‘Stay Fit’ is a single user application that uses various technologies to track the activities of user during workout. It runs on iOS device only and allow user to connect with any BLE device that implements HRM profile. It requires various capabilities of device to perform this task. The application uses GPS to track user’s location and uses BLE to connect with HRM.

As this is a single user application at a time only one user will be able to use services offered by this application. Though this application can be reset to allow the new user to start tracking his/her fitness but the data of old user will not be available after resetting.

The application also works without connecting any HRM device or without using location services, but in those cases the workout report generated by the application will not be accurate and will skip some of the attributes in calculation as in absence of peripheral or services.

The application can keep on running in background until user forcefully kills it, thus user can also continuously measure his/her heart rate and also can plot the recorded values on a graph, which can be very useful in investigation of many diseases. The application includes a very important feature of voice assistance during workout process to allow user to know his duration, goal completion and many other attributes, thus user can push his/her limits and complete a successful workout. Finally the user can also share the details of his/her workout on social network.

### **1.5 Organization**

Infostretch Solutions Private Limited.

## 1.6 Platform

iOS (6.0 and later)

## 1.7 Technologies Used

- **iOS Software Development Kit (SDK 7.0)**
  - **Graphics**
    - Core Graphics Framework
    - Foundation Framework
    - UI Kit Framework
  - **Animation**
    - Acceleration Framework
    - OpenGL ES Framework
    - Quartz Core Framework
    - Event Kit Framework
  - **Media**
    - Media Player Framework
    - AV Foundation Framework
    - Audio Toolbox Framework
  - **Map**
    - Map Kit Framework
  - **Location**
    - Core Location Framework
  - **BLE**
    - Core Bluetooth framework
  - **SQLite**
    - SQLite library for iOS
  - **Social**
    - Social Framework
- **Core Plot SDK**
  - Cocoa Touch Core Plot Framework

## 1.8 Database

SQLite 3.7.13

## 1.9 Methodology

Object Oriented System Development

## **1.10 Tools Used**

- XCode 5
- iPhone 6.1 and 7.0 Simulator
- Smart Git
- SQLite Explorer
- Iphone 4s, 5, 5s.

## **1.11 Project Execution Type**

Iterative Waterfall Model

## **1.12 Project Type**

Mobile Application

## **1.13 Project Duration**

9<sup>th</sup> December, 2013 to 29<sup>th</sup> March, 2014

## **Chapter 2**

### **About The System**

---

## 2.1 About ‘Stay Fit’

This mobile application can be used to track different parameters during workout. This app will be able to connect to BLE enabled Heart Rate Monitor. This monitor will continuously send user's Heart Rate data to application. This application will keep track of Heart Rate data along with user's Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map. User will be able to set his/her workout goals through this application. Application will notify user on completion of any goal. User will be able to view/share his/her daily/weekly/monthly workout statistic from app. This app will remind user for Workout if he/she has set any reminder. User will be able to Play music during workout.

‘Stay Fit’ uses a peripheral device, which has implemented the HRM (Heart Rate Monitor) profile of Bluetooth smart protocol. The application gets user’s heart rate in bps unit from the device and also records the location details and allows users to track his/her fitness, all in one app. In addition to that, the application keeps record of user’s daily activity and also shows a statistics of all workouts that user has undergone to. The application also uses GPS to accurately measure the distance covered and speed of user during workout. Over all, ‘Stay Fit’ helps user to stay fit.

## 2.2 Requirements for ‘Stay Fit’

### 2.2.1 Hardware Requirements

- **Development**

- Apple Macbook / Macmini
- iPhone 4s, 5, 5s
- Wahoo Blue HR

- **Implementation**

- RAM: 512MB min
- Processor: ARMv7, ARMv7s ARM64 or better
- iPhone 4s, 5, 5s
- GPS
- BLE

### 2.2.2 Software Requirements

- **Development**

- Xcode 5
- iOS simulators

- **Implementation**

- iOS 6.0 and later

## **2.3 FEASIBILITY STUDY**

An important outcome of the preliminary investigation is the determination that the system requested is feasible or not. There are three aspects in the feasibility study portion of the preliminary investigation.

The first study aspect is whether the current project is technically feasible i.e. can the work for the project be done with the current equipment, existing software technology, and available personnel? If a new technology is required then what is the likelihood that it can be developed?

The second study aspect probes whether the project is economically feasible i.e. are there sufficient benefits in creating the system to make the costs acceptable? Or, are the costs of not creating the system so great that the project must be undertaken.

The third study aspect probes whether the project is operationally feasible or not i.e. will the system be used if it is developed and implemented or whether there will be any resistance from the users that may undermine the possible application benefits.

### **2.3.1 Technical Feasibility**

This system is technically feasible because all required technologies are available today in market and the various guidelines are available on Apple's own website which is freely accessible and also includes various free sample applications which can be used as reference in development process. All other required tools like Xcode development tool, licensed version of OS X 10.8.2 (Mountain Lion) etc. and devices for testing the application are made available to us by the Organization.

### **2.3.2 Time Schedule Feasibility**

The job was assigned on 24<sup>th</sup> January 2011 and the time was sufficient as per the requirement of the project. The training completion date was 29<sup>th</sup> March 2012. So the project schedule was feasible. The project has complex features but the basic requirement can be satisfied within the allotted time period so the time development feasibility for this is satisfied.

### **2.3.3 Operational Feasibility**

The following areas have been probed to declare the proposed system as Operational feasible. This application can be used by any novice user, who have the basic knowledge of handling iOS Device.

### **2.3.4 Implementation Feasibility**

The project is feasible in its implementation. The entire development tools are available which can be used to make application run on an iOS device. We have a sound knowledge of Objective-C, iOS platform, OS X etc. and our team leader is

expert in iPhone mobile application development. Due to the mentioned factors, implementation of project was easy for us.

## 2.4 PROJECT PLANNING

### 2.4.1 Project Development Approach and Justification

For Project Development Iterative Waterfall Model is used.

#### Iterative Waterfall Model:

The Iterative waterfall model approach overcomes the problems associated with the waterfall model approach. If any difficulty or problem encounter in any phase may require going back to the previous phase and performing the required modifications and proceeds sequentially. This backtracking allows modifying any corrections or modifications required in the previous phase.

As illustrated in Fig 2.1, this model divides the cycle into the phases mentioned below:

1. Feasibility Study.
2. Requirements analysis and specification.
3. Design.
4. Coding and Unit Testing.
5. Integration and System Testing.
6. Maintenance.

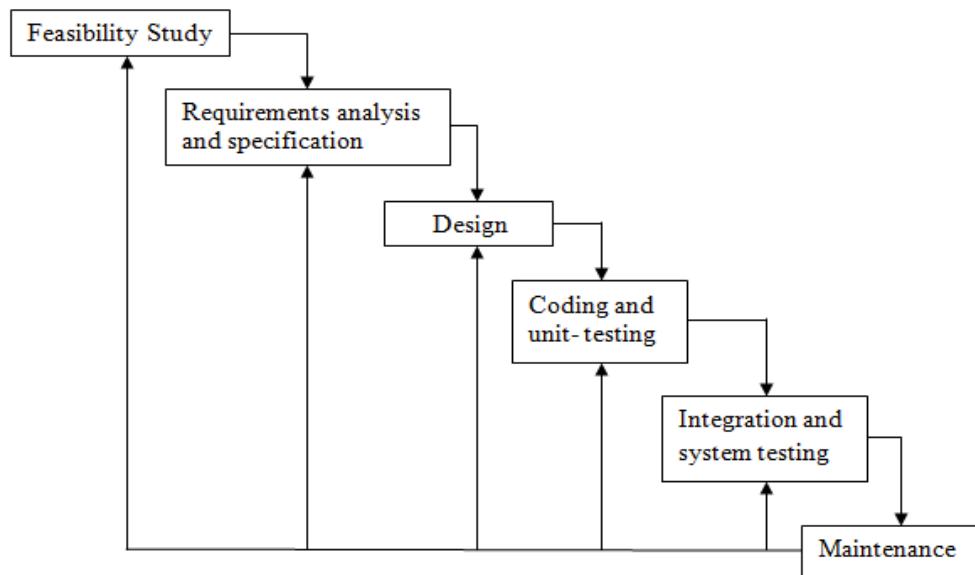


Figure 2.1: Iterative Waterfall Model

**Advantages of using Iterative Waterfall Model:**

- You are provided the chance to see the potential outcomes of every stage and make changes to areas of concern if necessary. This is one of the reasons that make the iterative model useful.
- Iterative development is more adjustable to changes as it considers each stage like a vital portion of the development cycle.
- The time spent on each successive interval may be lessened depending on how the last stage went and what knowledge was gained from past stages. The system therefore grows through adding new functionalities in the development part of all iterations.

**Disadvantages of using Iterative Waterfall Model:**

- When using the iterative model people working on the project can get stuck in a loop. Always finding problems than having to go back and design a fix, implement it, than test the system again and finding another problem can mean that the project can run over time and budget.
- Informal requests for improvement after each phase may lead to confusion and may also create scope creep, since user feedback following each phase may lead to increased customer demands. As users see the system develop, they may realize the potential of other system capabilities which would enhance their work, this can be an advantage as much as it can be a disadvantage.

**Justification:**

After feasibility study as the functional requirements were almost clear which were decided by our project lead, but UI related requirements were not clear. Here we have decomposed the system into modules. That is why we decided to use iterative waterfall model which is most suitable model here i.e. if we find any difficulty in coding and testing a modification in design can be done easily.

**Project Plan:**

After feasibility study as the functional requirements were almost clear which were decided by our project lead. After analyzing and thoroughly understanding the requirements of the application we planned the project.

In Design phase OO Design Technique is used. MVC (Model-View-Controller) architecture is used for ‘Stay Fit’ Mobile Application. Here we have decomposed the system into modules. Also the internals of the individual modules are designed in greater details. Coding and Unit Testing phase is required to translate the software design into source code. Also during this phase each module is unit tested to determine the correct working of all the individual modules. Integration and System Testing phase consists of the integration of the modules in a planned manner. Here during each integration step we have tested the partially integrated system. Finally, when all the modules were successfully integrated and tested, system testing was carried out successfully.

### 2.4.2 Milestones and Deliverables

Timely directions are always required to run a project successfully. Milestones tell the developers how far he has reached and also tell him what things are still left and how to fulfill them. Milestones may be the short report of achievement in project activity that are used by the project manager to check project progress but which are not delivered to the Clients. The deliverables are the project results that are provided to the customer. It is usually delivered at the end of some major project phases.

Table 2.1: Milestones and Deliverables

MILESTONES	DELIVERABLES	PURPOSE
Software Installation and Understanding of Technology	Had complete knowledge of OS X 10.8.2 (Mountain Lion) and its features.	To be familiar with OS X
System feasibility study, Requirement and Analysis	<ul style="list-style-type: none"> <li>• Requirement Gathering and analysis.</li> <li>• Functional Specifications</li> <li>• Non Functional Specifications</li> </ul>	It gives exact understanding of the user's requirements.
System Design	<ul style="list-style-type: none"> <li>• Use Case Diagram</li> <li>• Class Diagram</li> <li>• Sequence Diagram</li> <li>• State Diagram</li> <li>• ER Diagram</li> <li>• Database Design</li> <li>• Component Diagram</li> </ul>	It gives the logical structure that describes the system.
Coding and Unit testing and corrections if any	Individually Tested and Functional Modules.	It gives the required module
Integration and System Testing	The output obtained for the required functionality after implementing and doing various types of testing	Integrated System is ready

### 2.4.3 Roles and Responsibilities

INFOSTRETCH proposes the following Project Management structure for the successful development/customization, implementation and maintenance of the project.

The following is the recommended Project Management structure for the execution of the project.

#### Project Team Structure:

INFOSTRETCH proposes the following team structure for the successful implementation of the project. The INFOSTRETCH team will be headed by a Project Manager and will be supported by Project Leaders/Module Leaders and Team Members and Trainees.

INFOSTRETCH proposes a multi-disciplinary team for the project. INFOSTRETCH would draw in knowledge from its pool of resources with expertise ranging from business domain to the systems administration, systems architect, and Networking and Internet technology. These experts will act as consultants to the INFOSTRETCH Mobile project team. Other enabling groups within INFOSTRETCH such as Quality Assurance, Manpower Allocation Task Committee, Operations and Training have important role in the successful execution of the project.

- **Project Manager**

The responsibilities of the Project Manager are:

- Overall management and functioning of the web portal project.
- Project Planning.
- Monitoring the progress & status of the project on a regular basis.
- Managing the flow of work.
- Change Management: Identifying and evaluating the changes to the scope and content of work during the project, which may impact the original budget and schedule.
- Ensuring the quality of the deliverables.

- **Project Leader**

The responsibilities of the Project Leader are:

- Planning the application customization/development work in the project.
- Monitoring the day-to-day status of application development/customization activities in project.
- Planning and scheduling of enhancement to system in co-ordination with Project Manager.
- Ensuring the quality of the deliverables.
- Obtain clarifications necessary from the Project Management team regarding Functionality, GUI design etc over telephone (teleconference) or e-mail.
- Manage the flow of work within his/her team and allocate work to team members.
- Monitor the Quality of the deliverables being created.
- Reporting status to the Project Manager on a weekly basis.

- **Module Leader**

The responsibilities of a Module Leader include:

- Responsible for the work related to his/her group
- Allocate work to his module members
- Creation of SRS, Creation of the Design Document and provide required support to the project team in the development of the system
- Co-ordinate testing

- Perform Defect analysis and take corrective actions
- Ensure all the documentation with respect to the module is complete and up to date
- Report on a weekly basis the status of the module to the Project leader
- **Team member**

The responsibilities of a team member include:

- Carrying out the work allocated
- Report on a weekly basis to Project Leader

#### **2.4.4 Group Dependencies**

Module Leaders will report the progress of work in their respective modules to the Project Leader on a weekly basis. The Project Leaders will in turn submit weekly status reports to the concerned Project Managers.

The status report will contain:

- Activities completed since last report
- Activities planned until the next report
- Progress against plan
- Deliverables to CLIENT since last report
- Deliverables due to CLIENT in the next reporting period
- Outstanding issues

The Project Manager will monitor the progress of the activities against the plan and prepare a consolidated Monthly Report to Project Manager (PM) of CLIENT. This report will cover the project status and deviations against plan.

### 2.4.5 Project Scheduling Chart

Table 2.2: Project Scheduling Chart

Date Schedule list	9 dec to 23 jan	24 jan	26 jan to 27 jan	28 jan to 29 jan	30 jan to 14 feb	14 feb to 15 feb	16 feb to 26 feb	26 feb to 01 mar	02 mar	13 mar to 12 mar	16 mar to 29 mar
<b>Introduction to OS X, iOS and objective-C</b>	✓										
<b>Project Defination</b>		✓									
<b>Software requirement specification</b>			✓								
<b>Study about BLE</b>				✓							
<b>Design and implementation of BLE Connection</b>					✓						
<b>Testing of BLE Connection</b>						✓					
<b>Design and implementation Workout Module</b>							✓				
<b>Testing of Workout Module</b>								✓			
<b>Design and implementation Reminder Module</b>									✓		
<b>Testing of Reminder</b>										✓	
<b>Integration and system Testing</b>											✓

## **Chapter 3**

### System Analysis

---

### 3.1 Introduction

#### 3.1.1 Purpose

Obesity is the most prevalent, fatal, chronic, relapsing disorder of the 21st century. Obesity is a leading cause of world's mortality, morbidity, disability, healthcare utilization and healthcare costs. It is likely that the increase in obesity will strain our healthcare system with millions of additional cases of diabetes, heart disease and disability. Significantly, excess adiposity or obesity causes insulin secretion, which can cause insulin resistance that leads to type-2 diabetes.

Regular physical activity will help you to avoid these things. Physical activity is essential to prevent and reduce risks of many diseases and improve physical and mental health. It can even help you live longer—research from the American Journal of Preventative Medicine indicates that regular exercise can add up to five years to your life.

Since regularity in physical activity is very important, the development of the fitness application that can run in your mobile device is mandatory.

#### 3.1.2 Scope

This document provides details of the following:

1. Functional overview of the 'Stay Fit' application.
2. Broad functionalities within 'Stay Fit' application.
3. External and internal interfaces of the modules of 'Stay Fit'.

#### 3.1.3 How to Use this Document

All users of this document should carefully read it, and ensure the following:

- All requirements have been listed and detailed
- Functionality described in this document is clear and unambiguous.

#### 3.1.4 List Of Abbreviations

Table 3.1: List Of Abbreviation

Abbreviation/ Acronym	Description
BLE	Bluetooth Low Energy
API	Application Program Interface
SDK	Software Development Kit
GPS	Global Positioning System
GLES	Graphics Library Embedded System
HRM	Heart Rate Monitor
FPS	Frame per second
iOS	An operating system developed by Apple Inc. for mobile devices such as iPhone and iPad

## 3.2 Requirement Specifications.

A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behaviour of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation.

### 3.2.1 Functional Requirements

#### R1 User's Information

##### R1.1 Save User Profile

**Description:** Get user's general information, which is required in calculating of expended energy.

**Input:** Name  
DOB  
Gender  
Weight  
Height

**Processing:** The information will be used to calculate the calories burned during the particular workout session.

##### R1.2 Edit User Profile

**Description:** Edit user's general information, which is required in calculating of expended energy.

**Input:** New Values of following :  
Name  
DOB  
Gender  
Weight  
Height

**Processing:** The information will be saved and the new information will be used in processing.

### R1.3 Reset All Data

**Description:** The all data that is stored about the user is deleted and the application starts from the state when it was first installed. This way the application handles an event of change of user.

**Input:** User commands to reset the application.

**Processing:** The data that was saved for a particular user is deleted and now application is ready to be used by another user

## R2 Workout Process

### R2.1 Set Workout Goals

**Description:** This module allows user to set his/her workout goals.

**Input:** Distance (meters) or  
Duration (minutes) or  
Step Count (number of steps) or  
Energy Expended (kcal)

**Output:** Application will notify user on completion of any goal.  
Application will also show percentage of Goal achieved where user can frequently check.

**Processing:** The information about goal provided by user will be used to track the workout process and remind the user whenever necessary.

### R2.2 Workout Reminder

#### R2.2.1 Add New Workout Reminder

**Description:** User will be able to add a reminder for workout.

**Input:** Time of reminder  
Repeat sequence  
Keep Remind me up to date  
Alarm Configuration

**Output:** Application will remind user at the time of Workout.

**Processing:** The application reminders which are set by the user within the application are synchronized with the iOS application of reminders. So the alarms can be triggered at appropriate time by the iOS system itself.

**R2.2.2 Remove Workout Reminder**

**Description:** User will be able to delete an existing reminder.

**Input:** User selects a reminder to be deleted from a list.

**Output:** Application will delete a reminder.

**Processing:** The application removes the reminder from the default reminder tab of the application.

**R2.2.4 Edit Workout Reminder**

**Description:** User will be able to edit an existing reminder.

**Input:** New Attributes of the reminder.

**Output:** Application will modify a reminder.

**Processing:** The application saves new parameters of the reminder from the default reminder tab of the application.

**R2.2.5 Disable Workout Reminder**

**Description:** User will be able remove an alarm from reminder.

**Input:** Reminder to disable.

**Output:** Application will remove the alarm and synchronize.

**R2.2.6 Enable Workout Reminder**

**Description:** User will be able add an alarm to disabled reminder.

**Input:** Reminder to enable.

**Output:** Application will add an alarm and synchronize.

### R2.3 Start New Workout

**Description:** User will able to start new work out session either predefined or custom.

**Input:** The option of start workout is selected by user.

**Output:** User will able to view his/her workout attributes which will contain total workout duration, distance, step count, Path on Map, Avg. Speed, Energy Expended & percentage of goal achieved.

**Processing:** On starting a new workout, application will start collecting workout data like, Heart Rate, Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map.

### R2.4 Workout Report Generation

**Description:** User will be able to generate a report on the end of the workout session and also will be able to get the history of workout session he/she had previously undergone to.

**Input:** Duration during which the user had taken workout.

**Output:** Graph / report for workout duration, distance, step count, energy expended, heart rate for a day / week / month.

**Processing:** The information collected during a workout session is used to calculate the attributes of report.

## R3 HR Monitor Connection

**Description:** The user will be able to connect BLE peripheral defined according to HR profile of Bluetooth 4.0.

### R3.1 Discover BLE HR peripheral devices

**Input:** User will select the option to discover peripheral devices which provide the HR monitoring services.

**Processing:** Application will discover nearby HR Monitor devices.

**Output:** List of BLE devices with HR profile.

### R3.2 Connect to BLE device

**Input:** User will select the device to which the connection has to be made over Bluetooth

**Processing:** Application will connect to the device selected by user.

### R3.2 Disconnect BLE device

**Input:** User will select the command to disconnect from currently connected device.

**Processing:** Application will disconnect BLE device and user will now be able to connect to other devices.

## R4 Continuous Hear Rate Monitoring

### R4.1 Record HR continuously

**Description:** User can opt for continuous Heart Rate monitoring through this application.

**Input:** The HR of user measured in BPS, given by BLE device.

**Processing:** If user opts for this option, application will track user's heart rate for whole day. (one record for each second, total of  $24*60*60$  HR records per day).

### R4.2 Plot HR record on Graph

**Input:** The duration for which user wants to see the HR measurement which are previously recorded.

**Output:** The graph in which the HR records are plotted.

**Processing:** A graph is plotted in which the time is on X axis and the HR is on Y axis. The graph is scaled in best resolution according to the duration for which the user wants to see the HR measurements.

## R5 Voice Assistance

**Description:** Application will periodically provide workout information through voice guidance.

**Processing:** On starting a new workout, application will start collecting workout data like, Heart Rate, Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map and time to time the application will give feedback to user in form of audio.

## R6 Music Player

**Description:** User will able to play music from device's music library during workout.

**Processing:** On starting a new workout, application will start Playing music which will be directly played from default music application of the device.

## R7 Social Network Sharing

**Description:** User will able share the details of his/her workout to facebook and twitter account.

**Preconditions:** The user must have a account configured with the settings app of the device.

**Processing:** The attributes of the workout will be shared on the preconfigured account of the user.

### 3.2.2 Non Functional Requirements

#### R1 Usability

##### R1.1 Ease of Use

- This application is very user friendly so any non technical user can also use this application very easily.
- The graphs, which show HR measurements in application, will be interactive. The user will be able to change the scale of graph to see more or less finer data.

##### R1.2 Required User Ability

The user should know the basic handling of iOS devices

#### R2 Reliability

##### R2.1 Error Handling

1. The application should not crash
2. The application should not have memory leak
3. If any error occurs during runtime, it should notify user about it

##### R2.2 Ease of Recovery

If any crashing or error occurs in application, it should restore the all information and configuration data about workout.

#### R3 Performance

##### R3.1 Response Time

- The application will be able to record the HR of user continuously (1 record per each second).
- The application will be able to save the location (co-ordinates) which is visited by user during workout session so that it can show it on map when requested by user.

##### R3.2 Resource Usage

The memory requirement of application should be as minimum as possible. Also, the memory occupied by application should not affect the performance of other applications on that mobile.

**R4 Supportability****R4.1 Ease of Installation**

The installation procedure of application should be made easy so any novice user can install the application to his mobile.

**R5 Accuracy**

- The location tracked for the user will be taken from GPS hardware of device, configured for best possible accuracy.
- The expended energy measured, will be the best approximation of the original kcal burned by user during workout.
- The current HR of user depends upon the BLE device, which sends data. Thus the application is not responsible if the data sent by the device is not correct.

### 3.3 Use Case Diagram

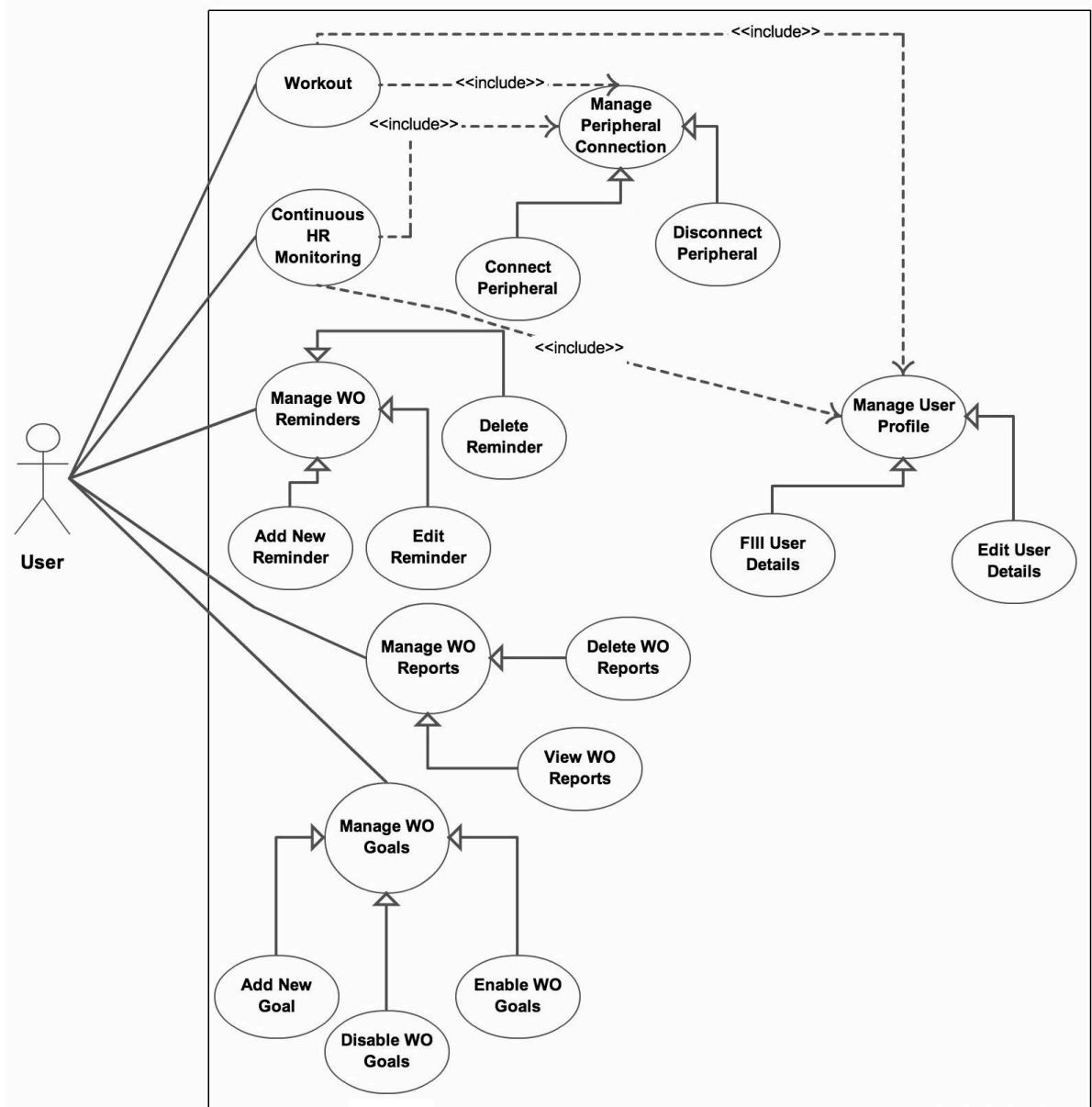


Figure 3.1: Use Case Diagram

### 3.4 Class Diagram

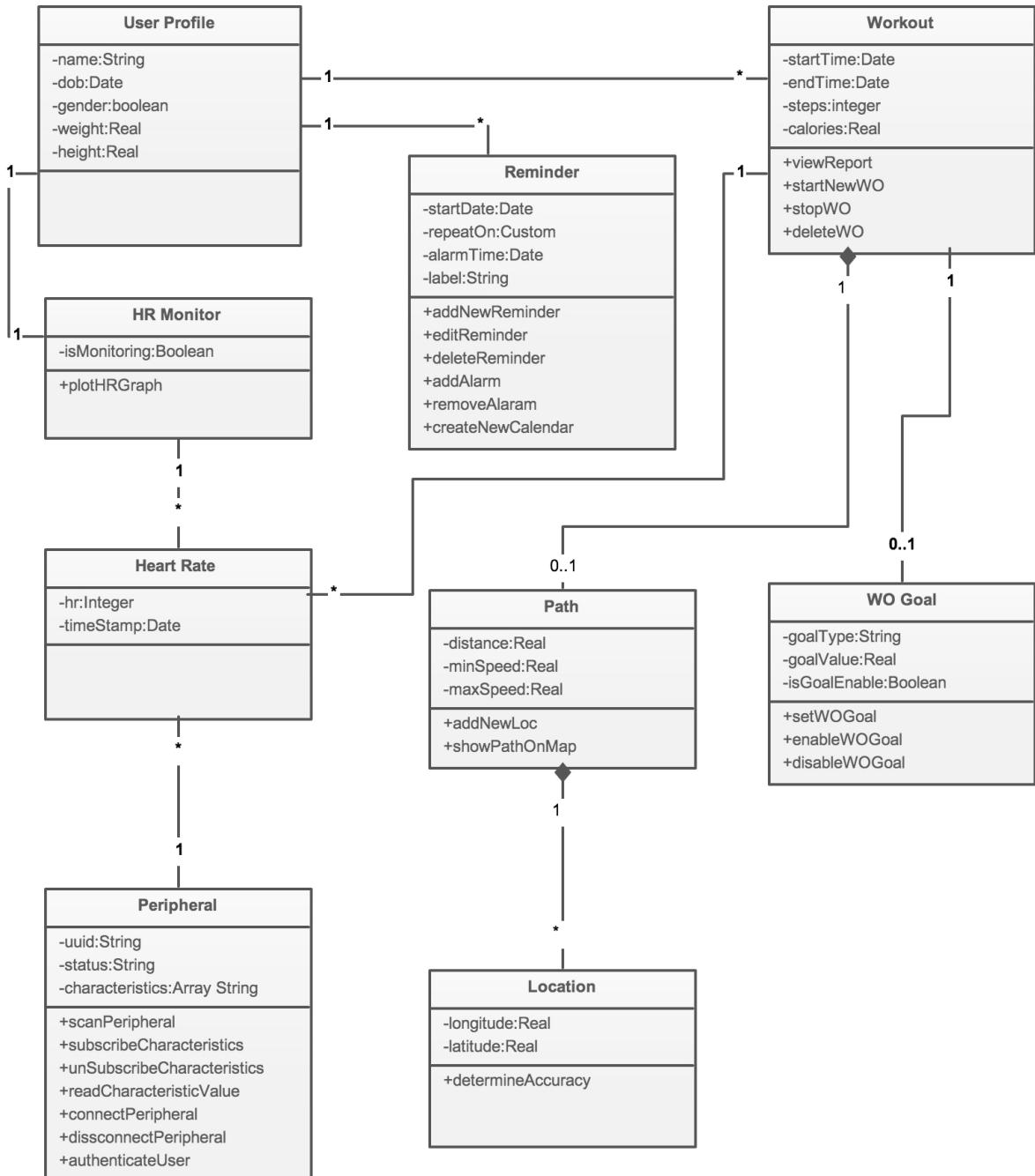


Figure 3.2: Class diagram

### 3.5 Sequence Diagrams

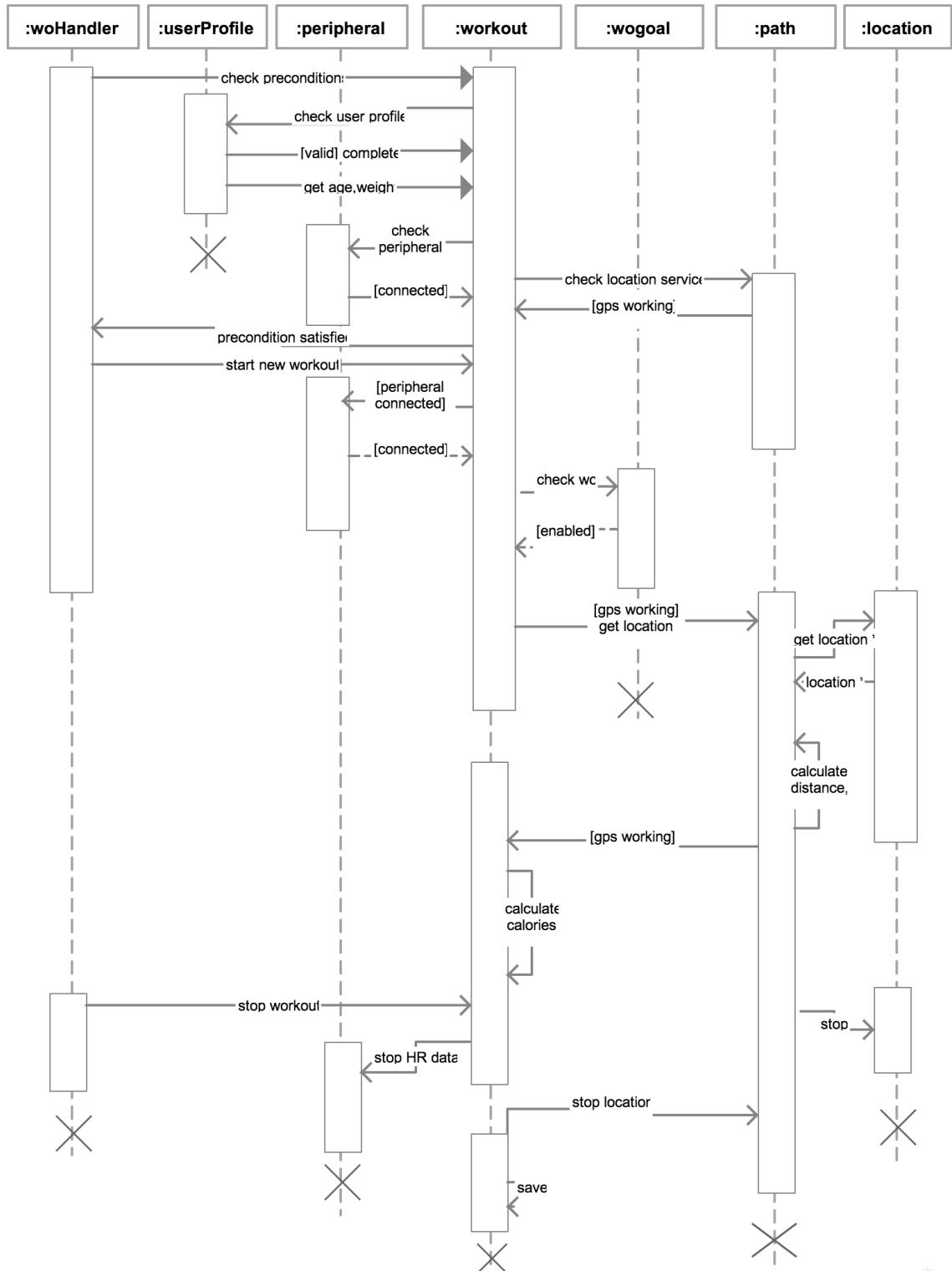


Figure 3.3: Sequence diagram for Start Workout

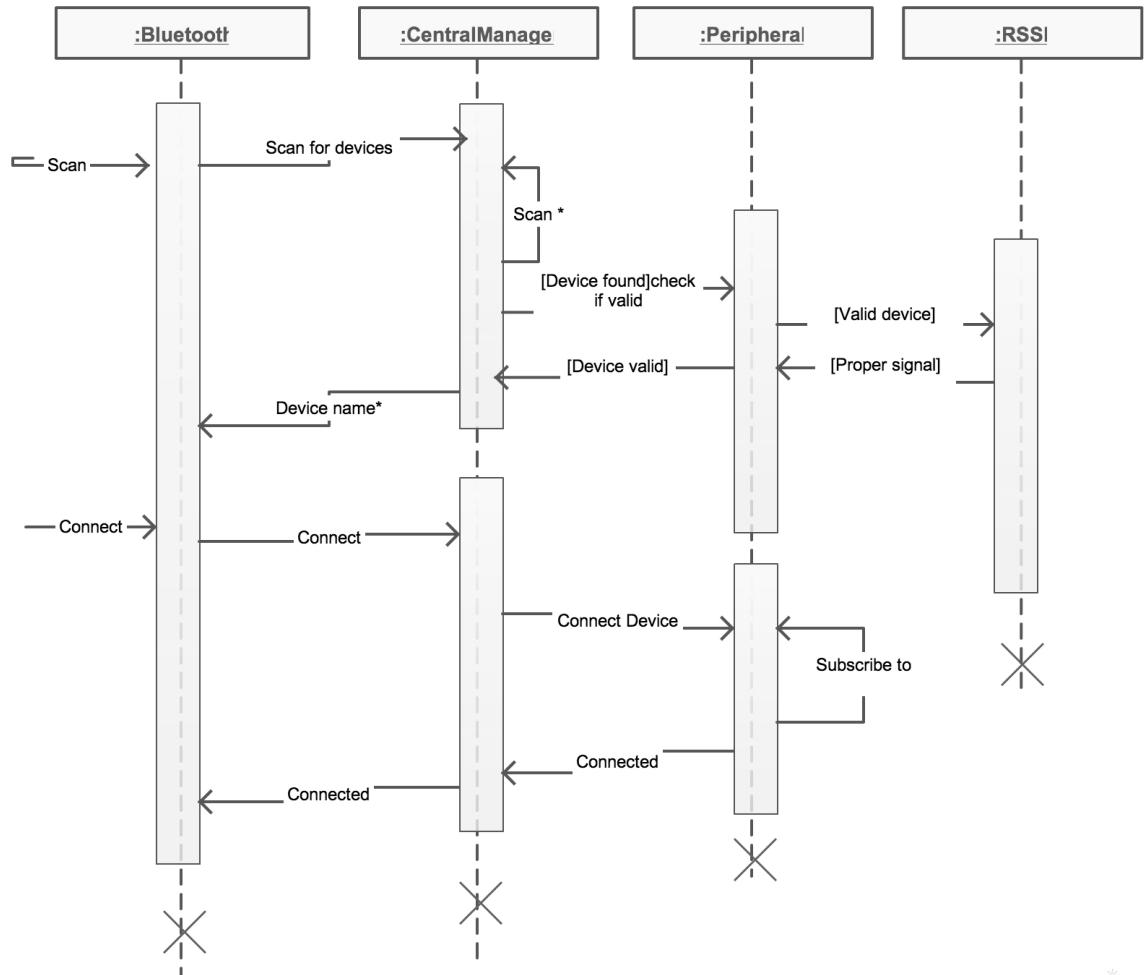


Figure 3.4: Sequence diagram for Connect BLE Device

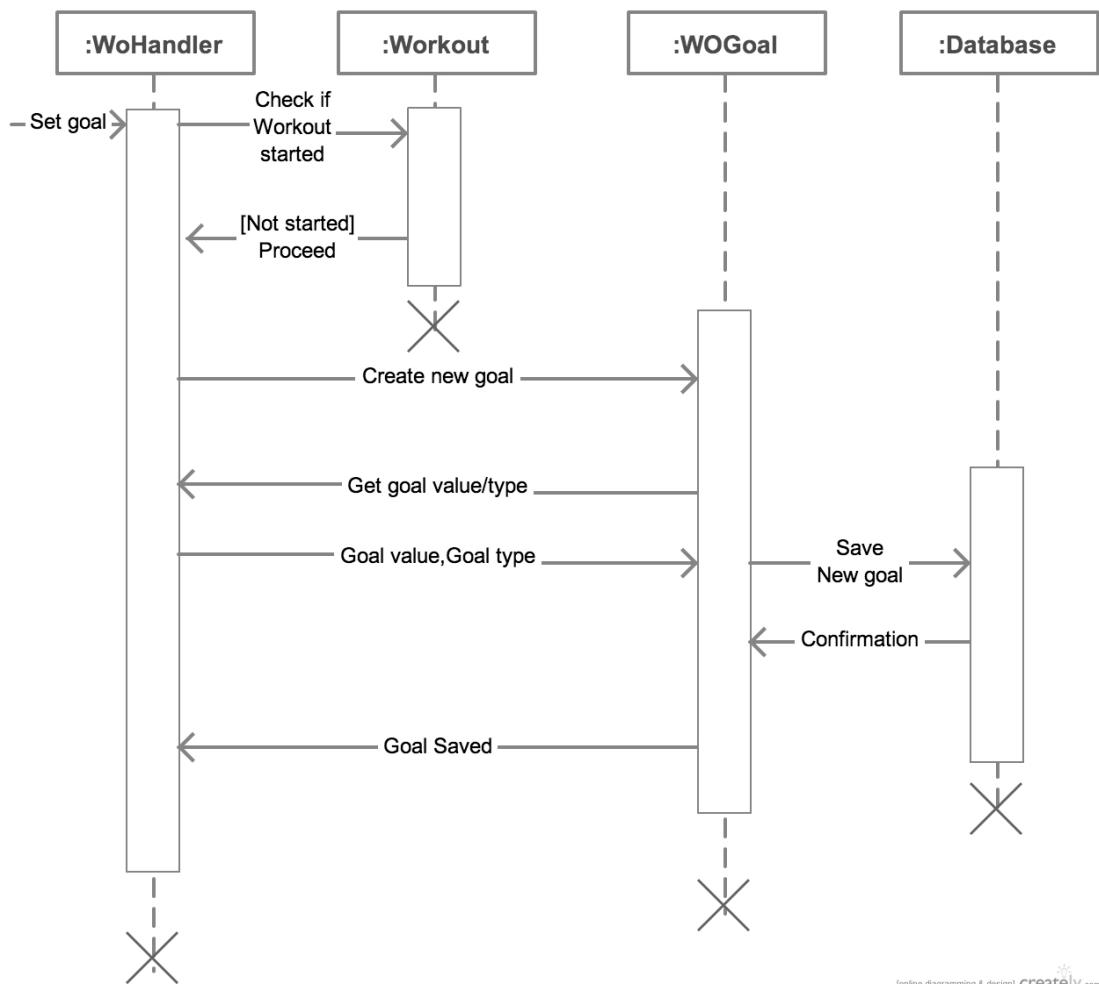


Figure 3.5: Sequence diagram for save workout goal

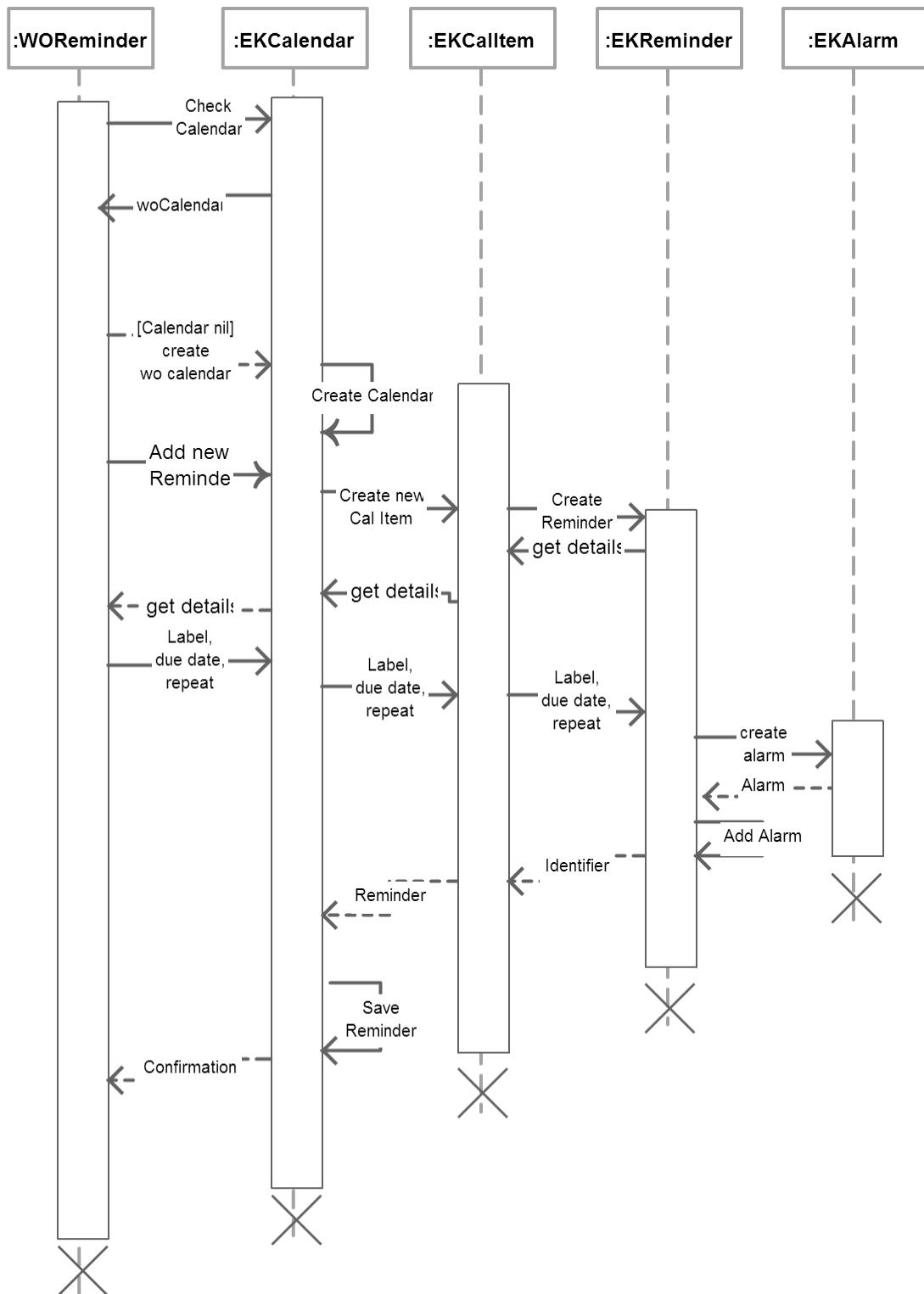


Figure 3.6: Sequence diagram for Adding new Reminder

### 3.6 State Diagrams

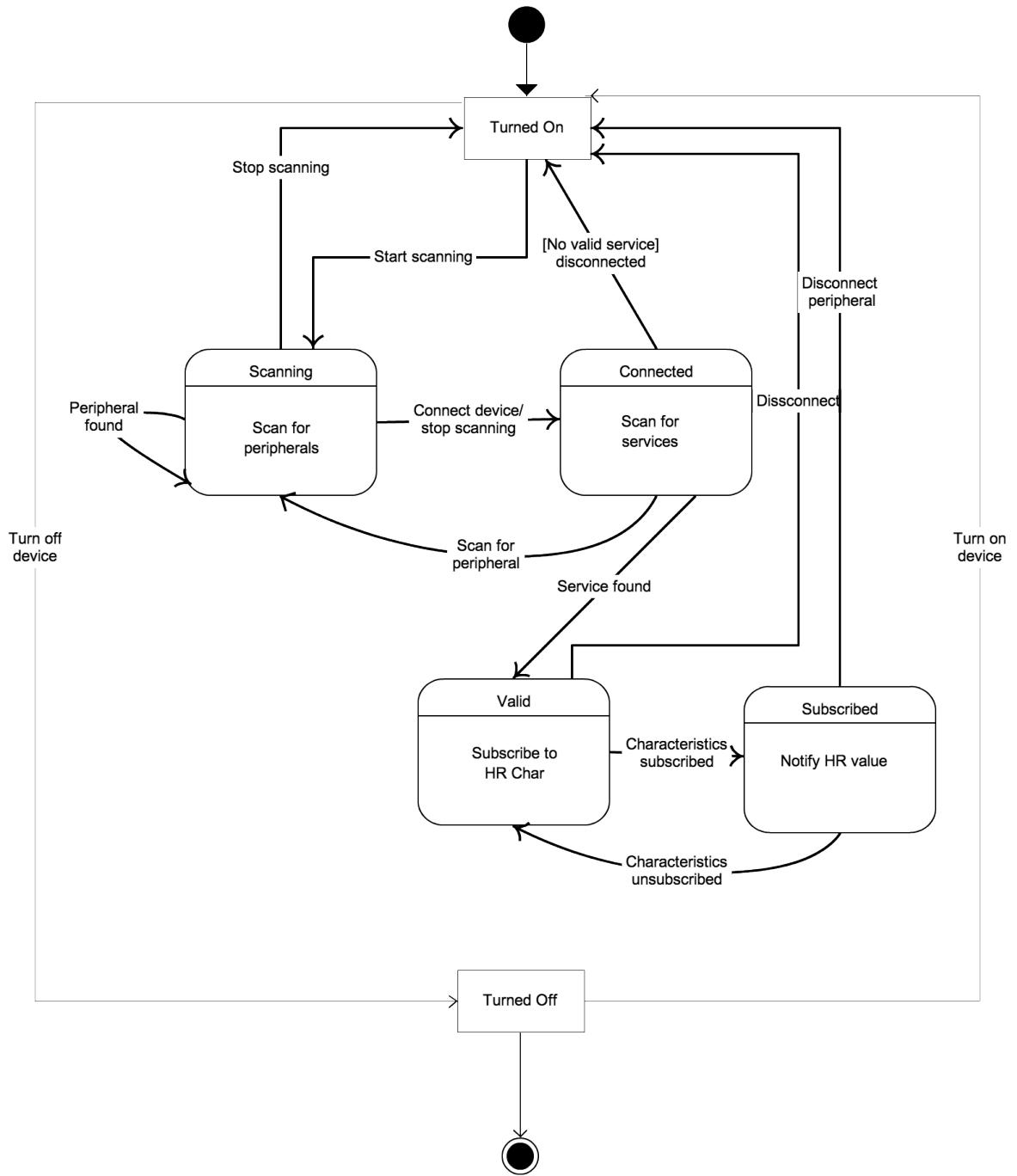


Figure 3.7: State diagram for Peripheral Object

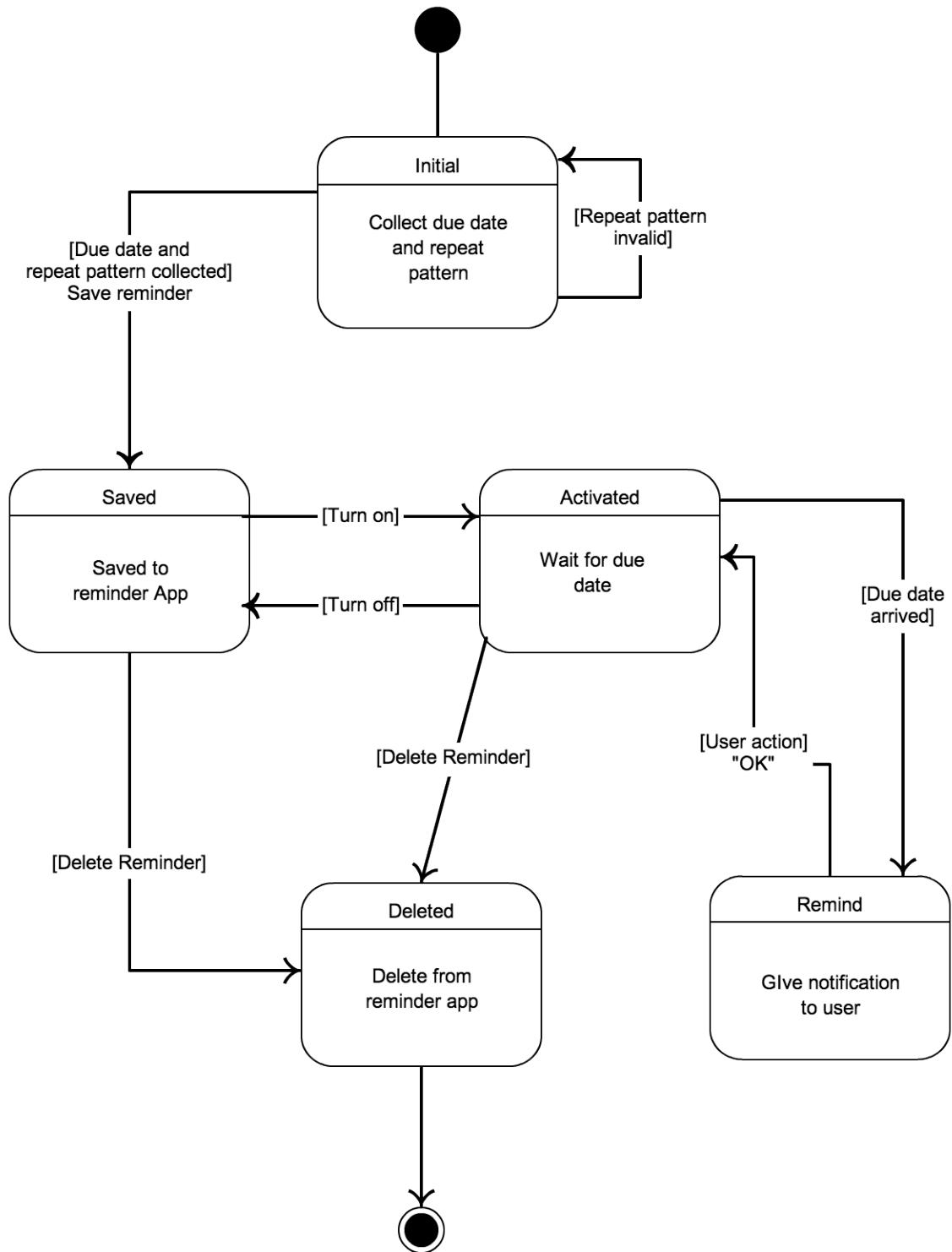


Figure 3.8: State diagram for Reminder Object

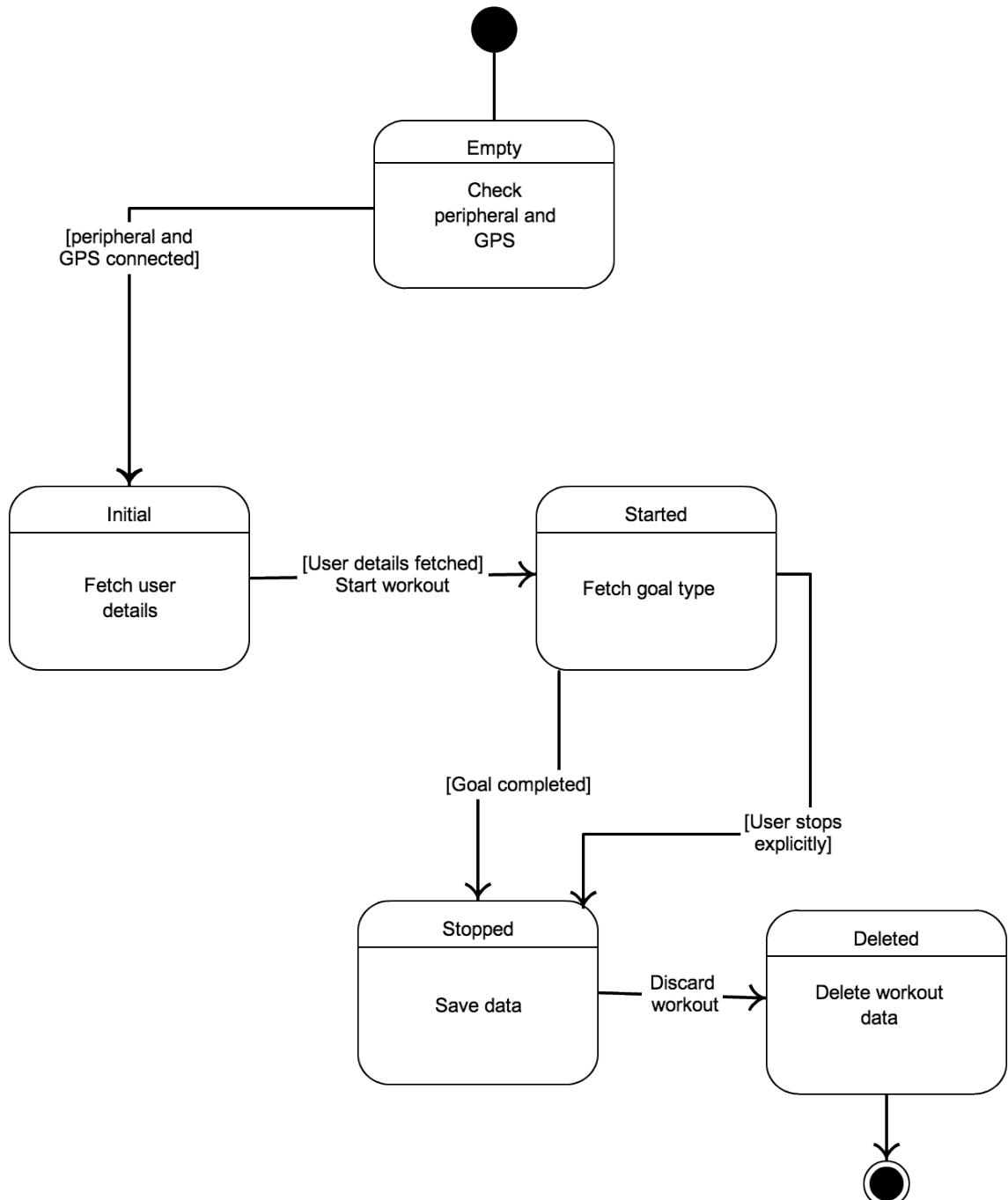


Figure 3.9: State diagram for Workout Object

### 3.7 Activity Diagram

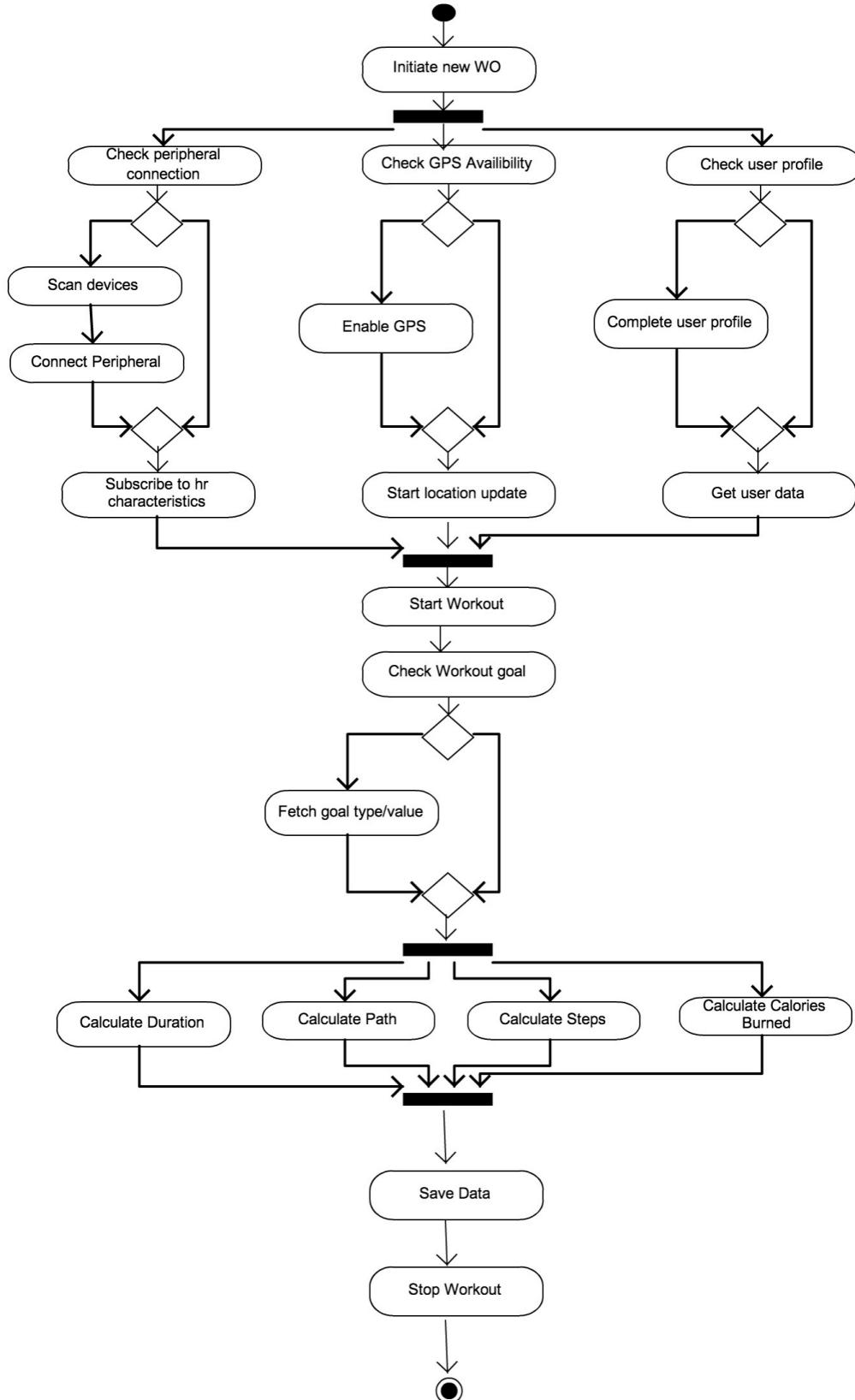


Figure 3.10: Activity diagram for Start Workout

## **Chapter 4**

### **Design**

---

## 4.1 Entity Relationship Diagram

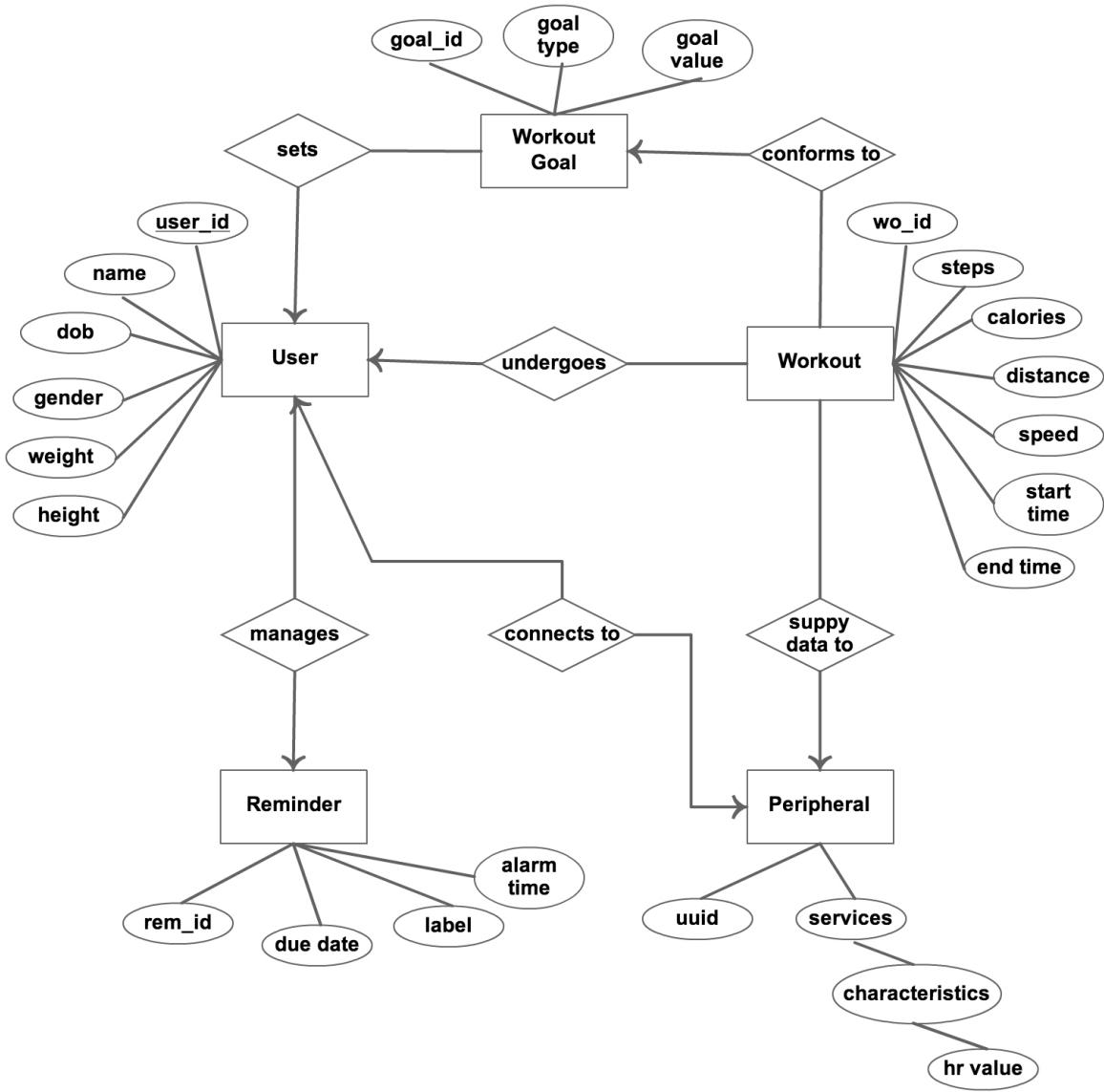


Figure 4.1: ER Diagram

## 4.2 Data Dictionary

**1 User\_Details** - This table contains all the important information of user.

Table 4.1 : User\_Details Schema

Column Name	Data Type	Nullable	Description
user_id	integer (auto_incr)	N	Unique user identification
name	text	N	User's name
dob	integer	N	Date of birth
gender	integer	N	Gender of user
height	real	N	User's height in cm
weight	real	N	User's weight in kg
hr_monitoring	integer	N	0 = disable , 1 = enable

**2 HR\_Details** - This table Stores Heart Rate data.

Table 4.2 : HR\_Details Schema

Column Name	Data Type	Nullable	Description
timestamp	integer	N	The time when the hr data was noticed
hr	integer	N	The heart rate value in bps

**3 Location\_Details** - This table Stores Location related data.

Table 4.3 : Location\_Details Schema

Column Name	Data Type	Nullable	Description
timestamp	integer	N	The time when the location data was aquired
latitude	real	N	The latitude value
longitude	real	N	The longitude value

**4 WO\_Goal\_Details** - This table Stores Workout Goal related data.

Table 4.4 : WO\_Goal\_Details Schema

Column Name	Data Type	Nullable	Description
wo_goal_id	integer (auto_incr)	N	Unique goal identification
goal_type	integer	N	1=miles, 2=calories, 3=duration
goal_value	real	N	Value of goal

**5 WO\_Details** - This table Stores Workout related data.

Table 4.5 : WO\_Details Schema

Column Name	Data Type	Nullable	Description
wo_id	integer (auto_incr)	N	Unique workout identification
start_timestamp	integer	N	The start time of workout
end_timestamp	integer	N	The end time of workout
steps	integer	N	Steps covered in workout
calories_burned	real	N	Calories burned during workout
min_speed	real	N	Min speed of running during workout
max_speed	real	N	Max speed of running during workout
distance	real	N	Distance covered in workout
goal_id	integer	N	The id of goal set for workout

### 4.3 Front-End Interface, Validations & Navigation Design

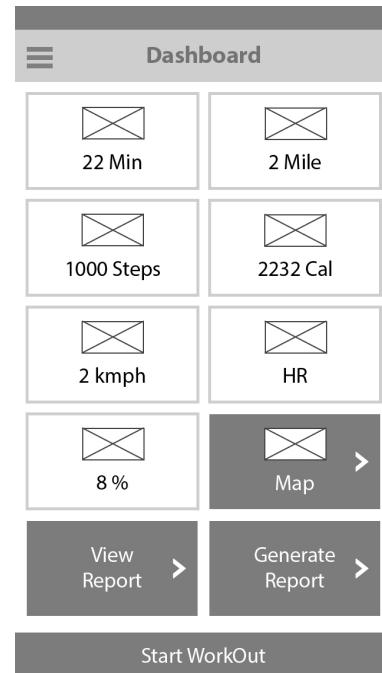


Figure 4.2: Dashboard Wireframe

A wireframe of a user profile registration form. At the top is a header bar with the text "User Profile". Below the header are several input fields: a text input for "Name" with a placeholder box, a date input for "D.O.B." with a calendar icon, a gender selection section with radio buttons for "Male" and "Female", a text input for "Weight", a dropdown menu for "Height", and a toggle switch for "Allow continuous heart rate monitoring". At the bottom is a dark grey "Register" button.

Figure 4.3: User Profile Wireframe

The wireframe for the 'WorkOut Goals' screen features a header bar with a back arrow and the title 'WorkOut Goals'. Below this is a section titled 'Set work out goal for the day'. It contains three input fields: 'Miles' with an input field, 'Calories' with an input field, and 'Duration' with an input field. At the bottom is a large 'Set' button.

Figure 4.4: Workout Goals Wireframe

The wireframe for the 'Connection Manager' screen features a header bar with a back arrow and the title 'Connection Manager'. Below this is a section titled 'Discover fitness device' with a refresh icon. A list of discovered devices is shown in a scrollable area, with 'abcd' currently selected. The list includes: abs, abcd, xyz, qwerty, aaa, and bbb.

Figure 4.5: Connection Manager Wireframe

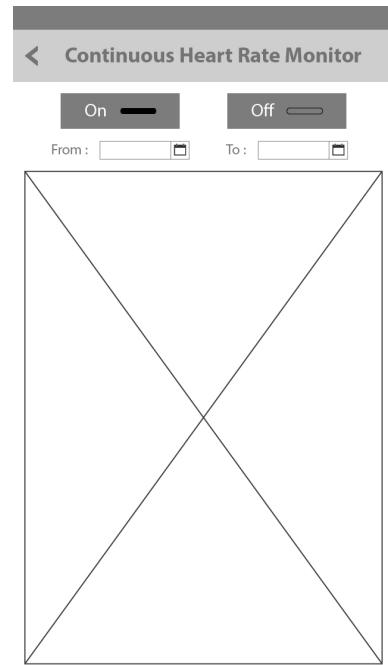


Figure 4.6: HR Monitor Wireframe

## **Chapter 5**

### Implementation

---

## Introduction:

Implementation includes all activities those take place to convert from old system to the new one. The new system may be totally new, replacing an existing manual or automated system, or it may be major modification to an existing system. In either case proper implementation is essential to provide a reliable system to meet organization requirements successful implementation but may not guarantee improvement in the organization using new system.

### 5.1 Implementation Environment

For Implementation we have used:

- XCode 5
- iPhone Simulator 6.0 or higher
- iOS SDK 7.0
- Smart Git

#### GUI based Vs. Non GUI

In GUI based application it is very easy for user to understand the overall content and the previous and next step of the action. Also GUI based application does not make as issues like navigation, ambiguities in data selection and entry.

GUIs have become the established alternative to traditional forms-based user interfaces. GUIs are the assumed user interface for virtually all the systems development using modern technologies. There are several reasons why GUIs have become so popular.

- GUIs provide the standard look and feel of a client operating system.
- GUIs are so flexible that they can be used in most application areas.
- The GUI provides seamless integration of custom and package applications.
- The user has a more natural interface to applications: user understanding is improved.

Table 5.1: Comparison between GUI and NON-GUI

<b>GUI</b>	<b>NON GUI</b>
In GUI based there is no hierarchical ordering of forms.	In Non GUI based applications, the forms are arranged in a hierarchical order.
The most obvious characteristic of GUI application it allows multiple windows to be displayed at the same time.	With Non GUI based application we can interact with one form at a time.
There are no constraints on the order in which user may enter data on screen.	In Non GUIs, the fields on the form have a predefined and unchangeable, the user may only access the fields in a certain order, regardless of whether any data has been entered into the form fields.

GUI introduces additional objects such as radio buttons, scrolling lists, check boxes and other graphics that may be displayed or directly manipulated.	Non GUI based do not have such objects.
---	---

Table 5.2 Comparison between Single user and Multi User

<b>Single User</b>	<b>Multi User</b>
The major disadvantage is that when one user is accessing a system other user cannot use it.	Multiple users can access the system paralleled.

Our system is mobile-app GUI and single-user system so no parallel access of the system will be there. In our system all connections with database is not done by the process of designing. There is proper and efficient coding for connection with database.

## 5.2 Technology and Literature Review

### 5.2.1 XCode 5

**Xcode** is an Integrated Development Environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. First released in 2003, the latest stable release is version 5 and is available via the Mac App Store free of charge for Mac OS X Lion and OS X Mountain Lion users. Registered developers can download preview releases and previous versions of the suite through the Apple Developer website.

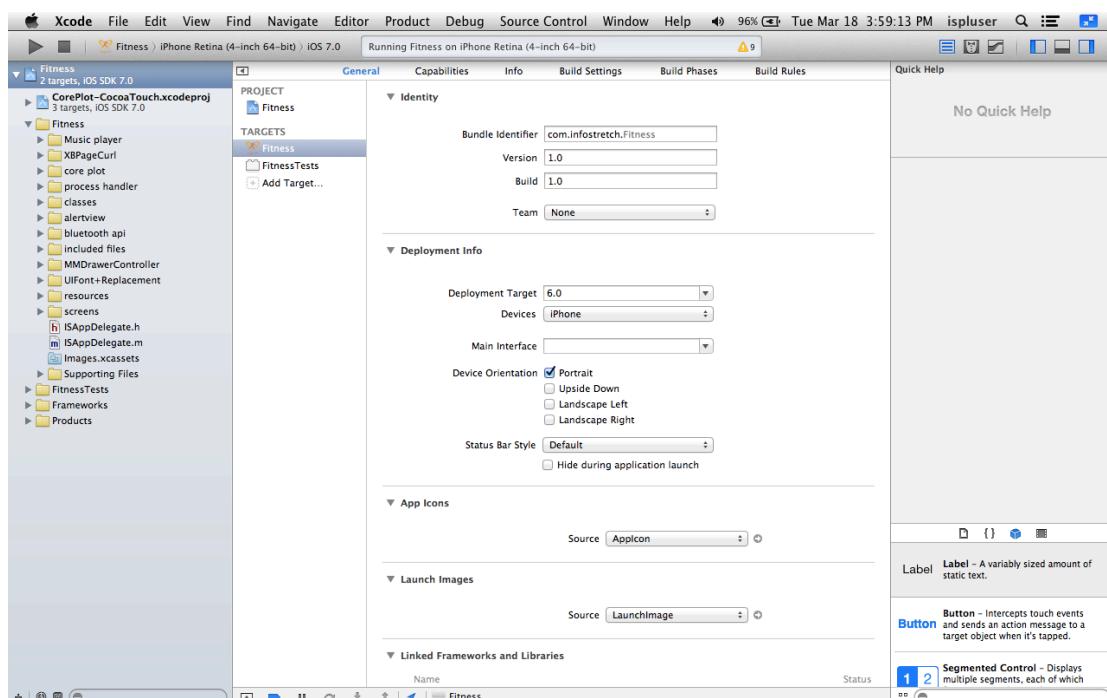


Figure 5.1: Xcode 5 Screenshot

The main application of the suite is the integrated development environment (IDE), also named Xcode. The Xcode suite also includes most of Apple's developer documentation, and built-in Interface Builder, an application used to construct graphical user interfaces.

### 5.2.2 iOS

**iOS** (previously **iPhone OS**) is a mobile operating system developed and distributed by Apple Inc. Originally released in 2007 for the iPhone and iPod Touch, it has been extended to support other Apple devices such as the iPad and Apple TV. Unlike Microsoft's Windows Phone (C++) and Google's Android, Apple does not license iOS for installation on non-Apple hardware. As of September 12, 2012, Apple's App Store contained more than 700,000 iOS applications, which have collectively been downloaded more than 30 billion times. In June 2012, it accounted for 65% of mobile web data consumption (including use on both the iPod Touch and the iPad). At the half of 2012, there were 410 million devices activated.

The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. Interface control elements consist of sliders, switches, and buttons. Interaction with the OS includes gestures such as swipe, tap, pinch, and reverse pinch, all of which have specific definitions within the context of the iOS operating system and its multi-touch interface. Internal accelerometers are used by some applications to respond to shaking the device (one common result is the undo command) or rotating it in three dimensions (one common result is switching from portrait to landscape mode).

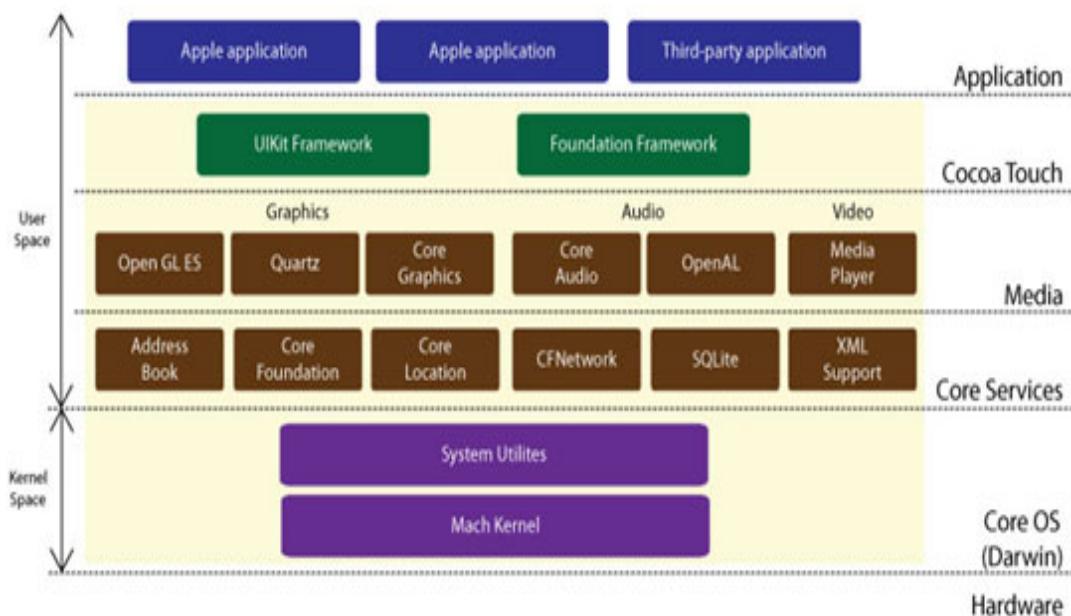


Figure 5.2: iOS structure

iOS is derived from OS X, with which it shares the Darwin foundation. iOS is Apple's mobile version of the OS X operating system used on Apple computers.

In iOS, there are four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The current version of the operating system (iOS 6.1.2) dedicates 1-1.5 GB of the device's flash memory for the system partition, using roughly 800 MB of that partition (varying by model) for iOS itself. iOS currently runs on iPhone, iPad, iPod touch and Apple TV.

### 5.3 Modules Specification

The ‘Stay Fit’ Mobile Application uses objective c classes along with xib files for GUI design. It also uses xml and plist files for implementation. It is stand-alone application and is not dependent on any web service.

Module performs following tasks :

- Set Workout Goals
- Set Workout Reminder
- HR Monitor Connection
- Continuous Hear Rate Monitoring
- Undergo Workout
- Workout Report
- Music Player
- Voice Assistance

#### 5.3.1 User Profile

This module deals with saving the user details into the application. It includes following details: Name, DOB, Gender, Weight and Height. The information will be used to calculate the calories burned during the particular workout session. This module also takes care of editing the user information and provides the other modules the details of user. It also resets and deletes all the data on user request.

#### 5.3.2 Workout

This module incorporate the handling code for a workout and requires to use services of all other modules. It communicates with other modules time to time and collects the information about user activity. It includes following task:

- Set workout goals.
- Start new work out session either predefined or custom.
- Generate a report on the end of the workout session and also get the history of workout session.

### 5.3.3 Reminder

This module interacts with the native reminder application to manage custom reminder that would be directly accessible from the application. This module provides the following functionality.

- Add a reminder for workout.
- Delete an existing reminder.
- Edit an existing reminder.
- Remove an alarm from reminder.

### 5.3.4 BLE Connection

This module uses the core Bluetooth API and provides the support to connect and maintain the data state while data exchange. It provides the following functionality.

- Connect BLE peripheral defined according to HR profile of Bluetooth 4.0.
- Discover peripheral devices which provide the HR monitoring services
- Disconnect from currently connected device.

### 5.3.5 HR Distributor

This module records the HR data and provides it to other modules as needed. This module is responsible for saving the heart rate data returned from BLE module. It also plots the graph as per the requirement of the user. It briefly includes following functionality:

- Continuous Heart Rate monitoring
- Plot HR graph

### 5.3.6 Voice Assistance

On starting a new workout, application will start collecting workout data like, Heart Rate, Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map and time to time the application will give feedback to user in form of audio. This module works in conjunction with the workout module. The workout module periodically passes the text to be spoken to this module and this converts the text data to speech.

### 5.3.7 Music Player

On starting a new workout, application will start Playing music which will be directly played from default music application of the device. This module communicates with the native music application of the iPhone and allows user to add the songs, which will be played during workout.

### 5.3.8 Social Network Sharing

Users will be able share the details of his/her workout to facebook and twitter account. The attributes of the workout will be shared on the preconfigured account of the user. This module also works in conjunction with workout module. The workout module provides the details of the workout and this module shares those on social network.

## 5.4 Coding Standards

Programming style is a set of rules or guidelines used when writing the source code for a computer program. It is often claimed that following a particular programming style will help programmers to read and understand source code conforming to the style, and help to avoid introducing errors.

### 5.4.1 General Principles

- **Clarity**

It is good to be both clear and brief as possible, but clarity shouldn't suffer because of brevity:

Example:

insertObject:atIndex:	Good.
insert:at:	Not clear; what is being inserted? what does "at" signify?

- **Consistency**

Try to use names consistently throughout the Cocoa programmatic interfaces. If you are unsure, browse the current header files or reference documentation for precedents.

Consistency is especially important when you have a class whose methods should take advantage of polymorphism. Methods that do the same thing in different classes should have the same name.

Example :

(NSInteger)tag	Defined in NSView, NSCell, NSControl.
- (void)setStringValue:(NSString *)	Defined in a number of Cocoa classes.

- **Class and protocol names**

The name of a class should contain a noun that clearly indicates what the class (or objects of the class) represent or do. The name should have an appropriate prefix (see "Prefixes"). The Foundation and application frameworks are full of examples; a

few are NSString, NSDate, NSScanner, NSApplication, UIApplication, NSButton, and UIButton.

Most *protocols* group related methods that aren't associated with any class in particular. This type of protocol should be named so that the protocol won't be confused with a class. A common convention is to use a gerund ("...ing") form:

NSLocking	Good.
NSLock	Poor (seems like a name for a class).

Some protocols group a number of unrelated methods (rather than create several separate small protocols). These protocols tend to be associated with a class that is the principal expression of the protocol. In these cases, the convention is to give the protocol the same name as the class.

An example of this sort of protocol is the NSObject protocol. This protocol groups methods that you can use to query any object about its position in the class hierarchy, to make it invoke specific methods, and to increment or decrement its reference count. Because the NSObject class provides the primary expression of these methods, the protocol is named after the class.

#### 5.4.2 Methods

- **Accessor Methods**

Accessor methods are those methods that set and return the value of a property of an object. They have certain recommended forms, depending on how the property is expressed:

If the property is expressed as a noun, the format is:

- (type)noun;
- (void)setNoun:(type)aNoun;

Example:

- (NSString *)title;
- (void)setTitle:(NSString *)aTitle;

If the property is expressed as an adjective, the format is:

- (BOOL)isAdjective;
- (void)setAdjective:(BOOL)flag;

Example:

```
- (BOOL)tableView:(NSTableView *)tableView shouldSelectRow:(int)row;
- (BOOL)application:(NSApplication *)sender openFile:(NSString *)filename;
- (BOOL)isEditable;
- (void)setEditable:(BOOL)flag;
```

- **Delegate Methods**

Delegate methods (or delegation methods) are those that an object invokes in its delegate (if the delegate implements them) when certain events occur. They have a distinctive form, which apply equally to methods invoked in an object's data source. Start the name by identifying the class of the object that's sending the message.

The class name omits the prefix and the first letter is in lowercase. A colon is affixed to the class name (the argument is a reference to the delegating object) unless the method has only one argument, the sender.

```
- (BOOL)applicationOpenUntitledFile:(NSApplication *)sender;
```

- **Method arguments**

There are a few general rules concerning the names of method arguments:

- As with methods, arguments start with a lowercase letter and the first letter of successive words are capitalized (for example, removeObject:(id)anObject).
- Don't use "pointer" or "ptr" in the name. Let the argument's type rather than its name declare whether it's a pointer.
- Avoid one- and two-letter names for arguments.
- Avoid abbreviations that save only a few letters.

### 5.4.3 Properties

A declared property effectively declares accessor methods for a property, and so conventions for naming a declared property are broadly the same as those for naming accessor methods. If the property is expressed as a noun or a verb, the format is:

`@property (...) type nounOrVerb;`

Example :

```
@property (strong) NSString *title;
```

```
@property (assign) BOOL showsAlpha;
```

If the name of a declared property is expressed as an adjective, however, the property name omits the “is” prefix but specifies the conventional name for the get accessor, for example:

```
@property (assign, getter=isEditable) BOOL editable;
```

In many cases, when you use a declared property you also synthesize a corresponding instance variable.

Make sure the name of the instance variable concisely describes the attribute stored. Usually, you should not access instance variables directly; instead you should use accessor methods (you do access instance variables directly in init and dealloc methods). To help to signal this, prefix instance variable names with an underscore (\_), for example:

```
@implementation MyClass {  
    BOOL _showsTitle;  
}
```

If you synthesize the instance variable using a declared property, specify the name of the instance variable in the @synthesize statement.

## **Chapter 6**

### Testing

---

## Introduction

Testing is the process carried out on software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications. Testing is advantageous in several ways. Firstly, the defects found help in the process of making the software reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the software is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

### 6.1 Testing Plan

The testing technique that is going to be used in the project is black box testing. In black box testing the expected inputs to the system are applied and only the outputs are checked.

The testing sub-process includes the following activities in a phase dependent manner:

- Create Test Plans.
- Create Test Specifications.
- Review Test Plans and Test Specifications.
- Conduct tests according to the Test Specifications, and log the defects.
- Fix defects, if any.
- When defects are fixed continue from activity.

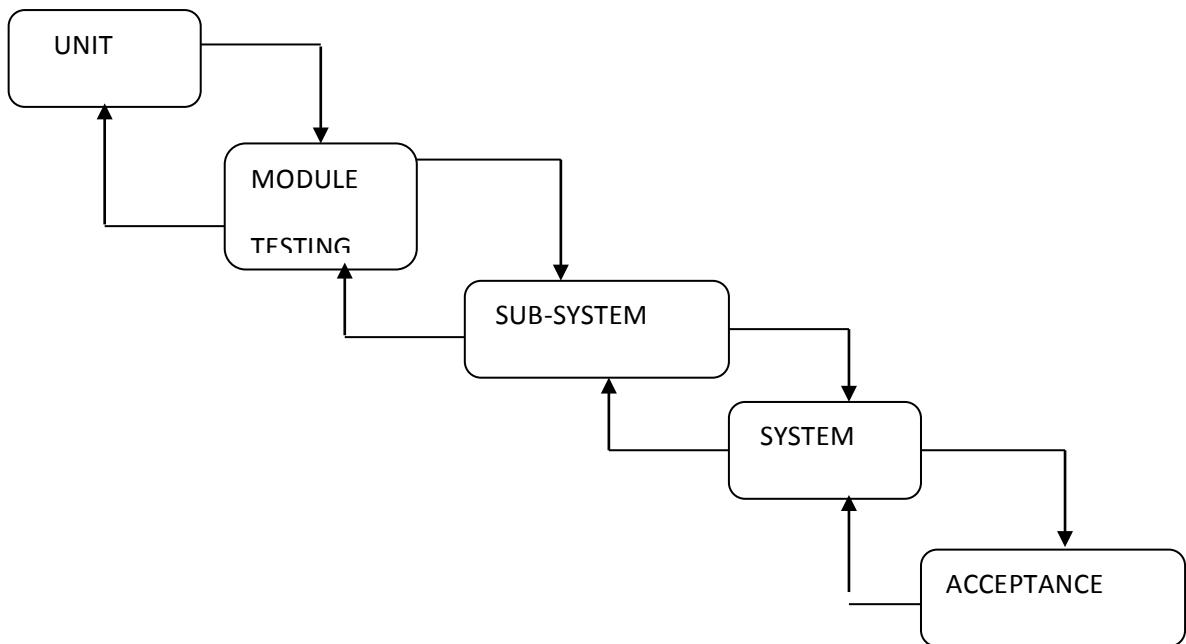


Figure 6.1 Testing Plan

## 6.2 Testing Strategy

The development process repeats this testing sub-process a number of times for the following phases.

- a) Unit Testing.
- b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

We have performed both Unit Testing and System Testing to detect and fix errors. A brief description of both is given below.

### 6.2.1 Unit Testing

#### Objective

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

**Input:** Unit Test Specifications

#### Testing Process

- Checking for availability of Code Walk-thru reports which have documented the existence of and conformance to coding standards.
- Review of Unit Test Specifications

Verify the Unit Test Specifications conform to the program specifications. Verify that all boundary and null data conditions are included.

## 6.3 TESTING METHODS

### Black-box and White-box Testing

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing.

For this purpose tools or compilers that can insert test probes into the programs can be used.

### Code Coverage

The way to make sure that you have got all the control flow covered is to cover all the paths in the program during the testing (via white-box testing). This implies that both branches are exercised for an ‘if’ statement, all branches are exercised for a case statement, the loop is taken once or multiple times as well as ignored for a while statement, and all components of complicated logical expressions are exercised. This is called Path Testing.

Branch Testing reports whether entire Boolean expression tested in control structures evaluated to both true and false.

Additionally it includes coverage of switch statement cases, exception handlers and interrupts handlers. Path testing includes branch testing as it considers all possible combination of individual branch conditions.

A simpler version is Statement Testing which determines if each statement in the program has been executed at least once. The coverage via Path Testing includes the coverage via Statement Testing. Since Path Testing is extremely comprehensive it is costly, hence a viable minimum should be measuring Statement Testing coverage.

### Performance Testing

Performance testing is design to test the runtime performance of the system within the context of the system. This test is performed at module level as well as at system level. Individual modules developed by Developers are tested for required performance.

## 6.4 Test Cases

A test case has a component that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly.

### 6.4.1 Save User Profile

Table 6.1: Save User Profile Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Profile tab	None	User profile screen shown	Pass
3	Tap on Save	None	Error: "Empty Fields"	Pass
4	Tap on Save	Empty DOB	Error: "Select DOB"	Pass
5	Tap on Save	Empty Name	Error: "Enter Name"	Pass
6	Tap on Save	Empty Height	Error: "Enter Height"	Pass
7	Tap on Save	Empty Weight	Error: "Enter Weight"	Pass
8	Tap on Save	Invalid Height	Error: "Enter Proper Height"	Pass
9	Tap on Save	Invalid Weight	Error: "Enter Proper Weight"	Pass
10	Tap on Save	Proper Details	Data Saved Successfully	Pass
11	Tap on Cancel	Proper Details	Error: "Enter Details to Proceed"	Pass

### 6.4.2 Edit User Profile

Table 6.2: Edit User Profile Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Profile tab	None	User profile screen shown	Pass
3	Tap on Save	None	Error: "Empty Fields"	Pass
4	Tap on Save	Empty DOB	Error: "Select DOB"	Pass

5	Tap on Save	Empty Name	Error: "Enter Name"	<b>Pass</b>
6	Tap on Save	Empty Height	Error: "Enter Height"	<b>Pass</b>
7	Tap on Save	Empty Weight	Error: "Enter Weight"	<b>Pass</b>
8	Tap on Save	Invalid Height	Error: "Enter Proper Height"	<b>Pass</b>
9	Tap on Save	Invalid Weight	Error: "Enter Proper Weight"	<b>Pass</b>
10	Tap on Save	Proper Details	Data Saved Successfully	<b>Pass</b>
11	Tap on Cancel	Proper Details	Changes Discarded	<b>Pass</b>

#### 6.4.3 Connect BLE Peripheral

Table 6.3: Connect BLE Peripheral Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Connection Manager tab	None	Connection Manager screen shown	<b>Pass</b>
3	Tap on Scan button.	None	Scanning Started Device Found	<b>Pass</b> <b>Pass</b>
4	Tap On Device Name	None	Device Connected	<b>Pass</b>

#### 6.4.4 Disconnect BLE Peripheral

Table 6.4: Disconnect BLE Peripheral Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Connection Manager tab	None	Connection Manager screen shown	<b>Pass</b>

3	Tap On Connected Device Name	None	Confirmation Asked	<b>Pass</b>
4	Tap On Yes	None	Device Disconnected	<b>Pass</b>

#### 6.4.5 Save Workout Goal

Table 6.5: Save Workout Goal Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Workout Goal tab	None	Workout Goal screen shown	<b>Pass</b>
3	Tap on Set	None	Error: "Empty Fields"	<b>Pass</b>
4	Tap on Set	Invalid Duration	Error: "Enter Proper Duration"	<b>Pass</b>
5	Tap on Set	Invalid Distance	Error: "Enter Proper Distance"	<b>Pass</b>
6	Tap on Set	Invalid Calories	Error: "Enter Proper Calories"	<b>Pass</b>
7	Tap on Set	Proper Details	Goal Saved Successfully	<b>Pass</b>

#### 6.4.6 Enable Workout Goal

Table 6.6: Enable Workout Goal Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Workout Goal tab	None	Workout Goal screen shown	<b>Pass</b>
3	Tap on Enable WO Goal	None	Error: "Select Goal Type"	<b>Pass</b>

4	Select Goal Type	Goal Value	Goal Selected	<b>Pass</b>
5	Tap on Enable WO Goal	None	Workout goal enabled	<b>Pass</b>

#### 6.4.7 Disable Workout Goal

Table 6.7: Disable Workout Goal Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Workout Goal tab	None	Workout Goal screen shown	<b>Pass</b>
3	Tap on Disable WO Goal	None	Workout goal disabled	<b>Pass</b>

#### 6.4.8 Play Music

Table 6.8: Play Music Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on Music Button	None	Music Panel Appears	<b>Pass</b>
2	Tap on Play button	None	Error: "Empty Playlist"	<b>Pass</b>
3	Tap on Add button	None.	Music Picker Displayed	<b>Pass</b>
4	Tap on Tracks	Tracks to be added	Tracks selected	<b>Pass</b>
5	Tap on Done	None	Tracks added to playlist	<b>Pass</b>
6	Tap on Play button	None	Music Played	<b>Pass</b>
7	Tap on Pause button	None	Music Paused	<b>Pass</b>
8	Tap on Next button	None	Next track played	<b>Pass</b>

9	Tap on Previous button	None	Previous track played	<b>Pass</b>
---	------------------------	------	-----------------------	-------------

#### 6.4.9 Delete Today's Workout Report

Table 6.9: Delete Today's Workout Report Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on View Report in Dashboard	None	Today's workout list displayed	<b>Pass</b>
2	Tap on workout report	None.	Workout details displayed	<b>Pass</b>
3	Tap on Delete button	None	Workout details deleted	<b>Pass</b>

#### 6.4.10 Delete Random Workout Report

Table 6.10: Delete Random Workout Report Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	<b>Pass</b>
2	Tap on Workout reports tab	None	All Workouts displayed	<b>Pass</b>
3	Tap on workout report	None	Workout details displayed	<b>Pass</b>
4	Tap on Delete button	None	Workout details deleted	<b>Pass</b>

### 6.4.11 Share Workout Report

Table 6.11: Share Workout Report Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Workout reports tab	None	All Workouts displayed	Pass
3	Tap on workout report	None	Workout details displayed	Pass
4	Tap on Share button	None	Share menu appears	Pass
5	Tap on Facebook button	None	Facebook Dialog appears	Pass
6	Tap on Post button	None	Status updated on Facebook	Pass

### 6.4.12 Share Workout Statistics

Table 6.12: Share Workout Statistics Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Statistics tab	None	All Workout Statistics displayed	Pass
3	Tap on Share button	None	Share menu appears	Pass
4	Tap on Twitter button	None	Twitter Dialog appears	Pass
5	Tap on Tweet button	None	Tweet sent	Pass

### 6.4.13 Plot HR

Table 6.13: Plot HR Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Heart Rate Monitor tab	None	Heart Rate Monitor displayed	Pass
3	Tap on From	From Date	Date picker appears	Pass
4	Tap on To	To Date	Date picker appears	Pass
5	Tap on Done	None	Plot Appears	Pass
6	Zoom in Plot	None	Units Changed	Pass

### 6.4.14 Add New Reminder

Table 6.14: Add New Reminder Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Workout Goals tab	None	Workout Goals screen shown	Pass
3	Tap on Workout Reminders tab	None	Workout Reminders screen shown	Pass
4	Tap on Add button	None	New Reminder screen appears	Pass
5	Tap on Save	Empty Title	Error: "Enter Title"	Pass
6	Tap on Save	Empty Alert	Error: "Select Alert"	Pass
7	Tap on Save	Empty Repeat	Error: "Select Repeat"	Pass
8	Tap on Save	Proper data	Reminder Saved Successfully	Pass

### 6.4.15 Edit Reminder

Table 6.15: Edit Reminder Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Workout Goals tab	None	Workout Goals screen shown	Pass
3	Tap on Workout Reminders tab	None	Workout Reminders screen shown	Pass
4	Tap on Reminder & edit button	None	Edit Reminder screen appears	Pass
5	Change Time	Start Time	Start Time changed	Pass
6	Tap on Save	Proper data	Reminder Updated Successfully	Pass

### 6.4.16 Delete Reminder

Table 6.16: Delete Reminder Test Case

No.	Steps	Input	Expected Results	Status
1	Tap on menu button	None	Menu shown	Pass
2	Tap on Workout Goals tab	None	Workout Goals screen shown	Pass
3	Tap on Workout Reminders tab	None	Workout Reminders screen shown	Pass
4	Tap on Reminder	None	Reminder Selected	Pass
5	Tap on Delete button	None	Conformation asked	Pass
6	Tap on Yes	None	Reminder Deleted Successfully	Pass

## **Chapter 7**

### User Manual

---

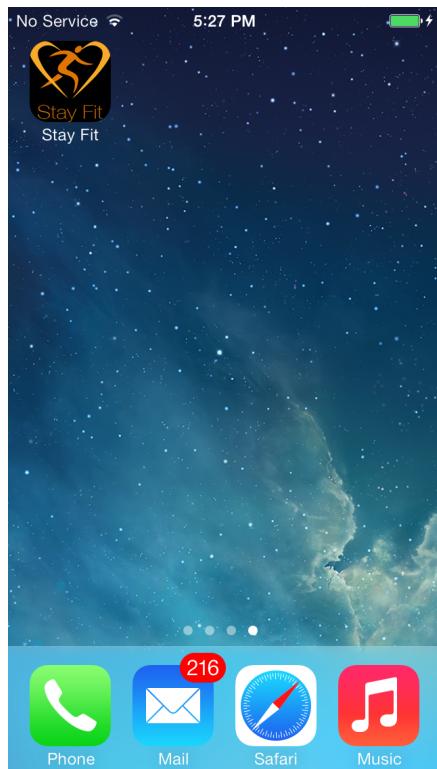
## Introduction

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system. It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies.

User guides are most commonly associated with electronic goods, computer hardware and software.

Our user guides contain both a written guide and the associated images. In the case of our application, it is usual to include screenshots of how the program should look. The language used is matched to the intended audience.

### 7.1 Splash Screen



#### I. 'Stay Fit' App Icon

This is the icon of 'Stay Fit' application.

Click on this icon to launch the application first time.

Figure: 7.1.1



Figure: 7.1.2

## II. ‘Stay Fit’ Launch Screen

This is the splash screen which will be displayed while the application is being launched.

The application will be launched in few seconds.

## 7.2 Registration

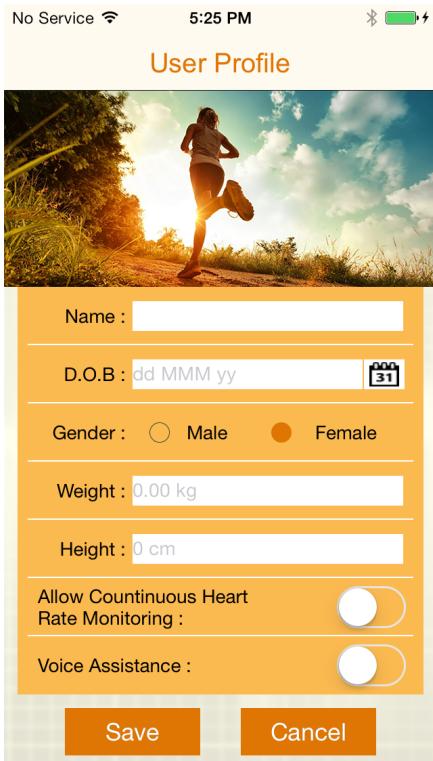


Figure: 7.2.1

## I. First Screen

This is the first screen that is displayed when an application is launched for first time.

Fill the details as following:

**Name:** Keypad

**Date of Birth:** Date Picker

**Gender:** Select Radio buttons

**Weight:** (kg)

**Height:** (cm)

### Note:

**Voice Assistance:** keep this option off if you don't want the device to give voice feedback during workout process.

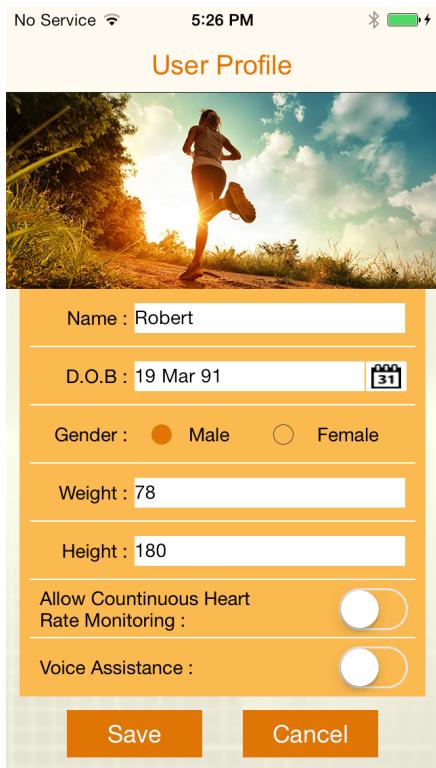


Figure: 7.2.2

## II. Sample Filled Profile

This is the sample of filled user profile without any errors.

### Note:

All the fields are mandatory.

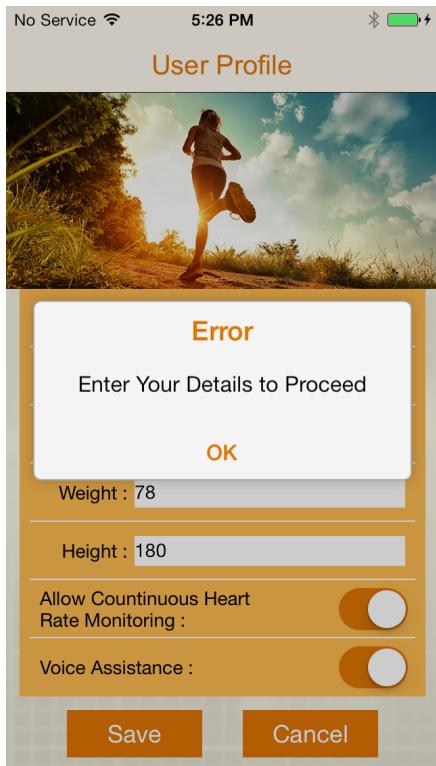


Figure: 7.2.3

## III. Incomplete Profile

After filling the details click on “Save” button to proceed.

At this stage if user get the error displayed on this screen, then any one of the field must be empty. Fill the data properly and proceed.

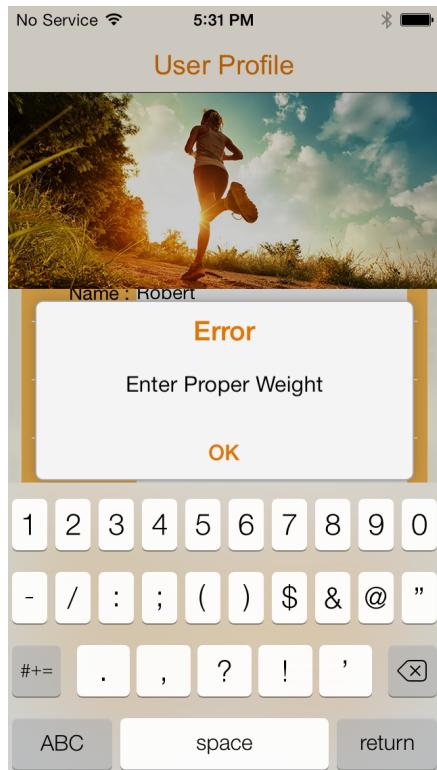


Figure: 7.2.4

#### IV. Validations

In User profile screen if user is seeing following error then some of the detail field are not filled properly.

Example:

Entering non numerical value in Weight field will give the error displayed when user saves the profile.

Check the appropriate field and reenter the data then click save to proceed.

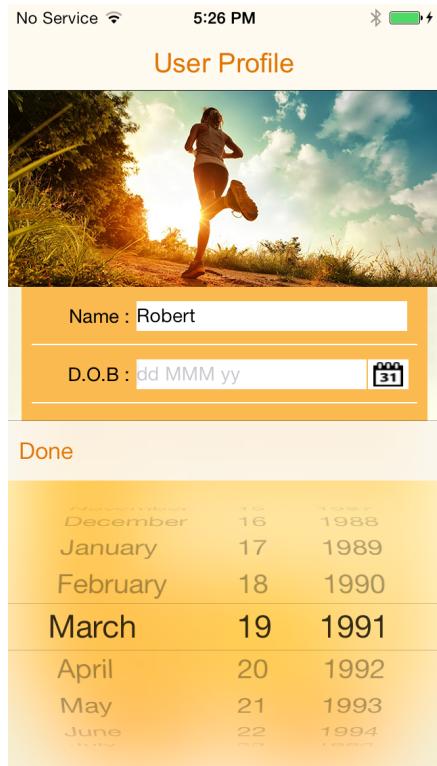


Figure: 7.2.5

#### V. D.O.B picker

Note:

After selecting the DOB field to fill, select the DOB from the picker and do not click on "Done" button until the picker is fully stabilized to a particular date.

Select the DOB properly as this information is mandatory for proper functioning of the app 'Stay Fit'.

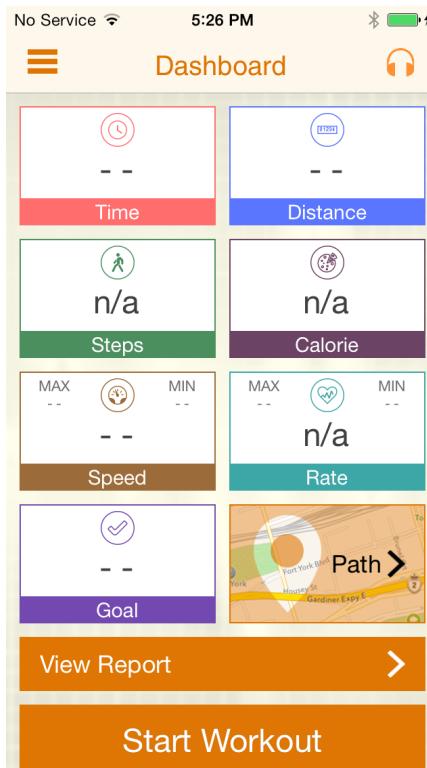


Figure: 7.2.6

## VI. Menu Button

This is a Dashboard screen, which will be displayed after the user profile is successfully saved.

The user can see the options in the “Menu” by clicking on the top-left button shown in the image.

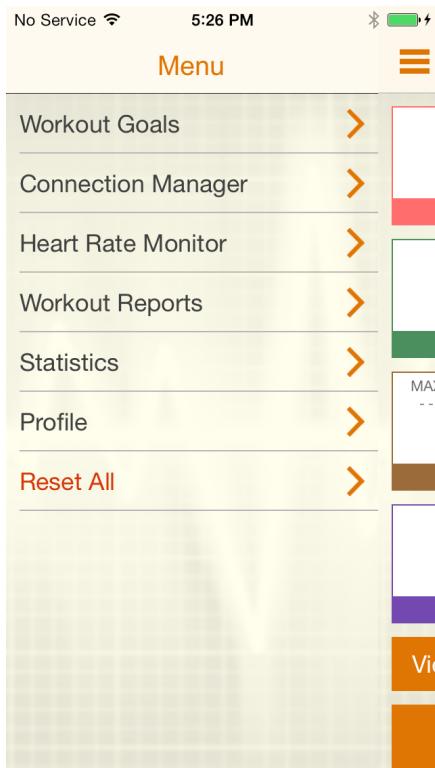


Figure: 7.2.7

## VII. Menu

This is the Menu from where the user will be able to navigate anywhere in the application.

The “Profile” tab is used to go to the “User Profile” screen from where the user can edit the details previously filled.

### 7.3 Connection Manager

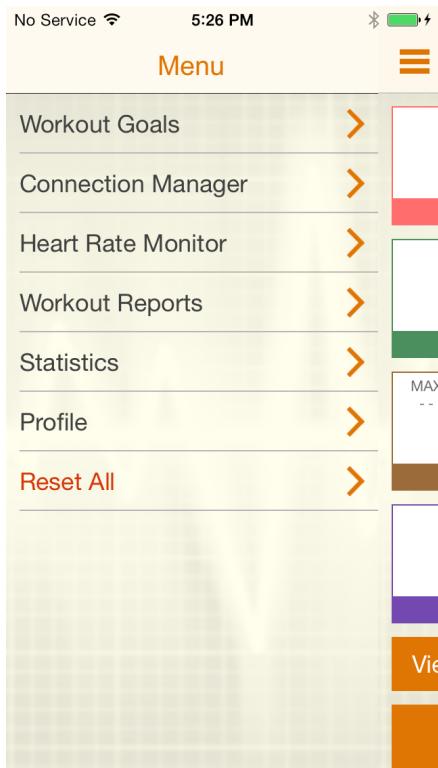


Figure: 7.3.1

#### I. Navigate to Connection Manager

This is the “Menu” which has a tab named “Connection Manager”. Click on the tab shown in the Figure to scan the BLE peripherals.



Figure: 7.3.2

#### II. Connection Manager

This screen allows a user to connect to the BLE peripheral.

Click on the button shown in the Figure to scan for the available devices.



Figure: 7.3.3

### III. Scanning

While scanning, all the available peripherals will be listed under “Available Devices”.

The Factory name of the device will help the user to select the proper device to connect.

Note:

If your device is not being displayed over here then follow the steps.

- i. Forcefully kill and restart ‘Stay Fit’.
- ii. Turn Bluetooth off then on.
- iii. Kill all other applications.
- iv. Reset the peripheral and check the batteries.



Figure: 7.3.4

### IV. Connect to Peripheral

Click on desired peripheral to connect to it. When connected a small dot will be displayed on the right side of the peripheral name as shown in the Figure.

## V. Disconnect Peripheral

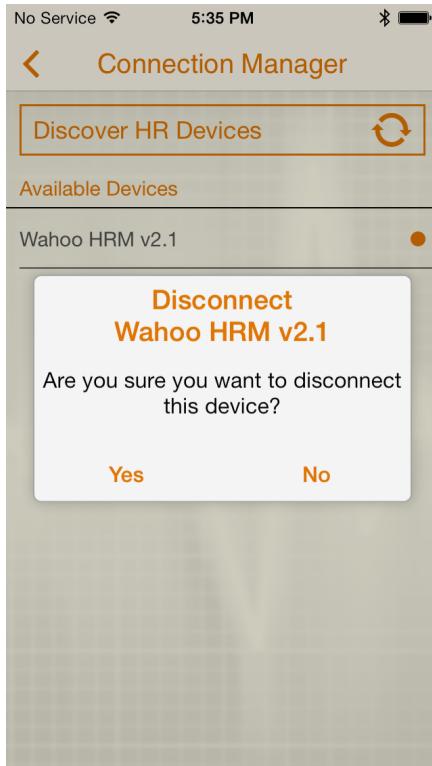


Figure: 7.3.5

To disconnect a device just tap on the tab with “orange circle”. A user will be asked for confirmation as shown in the Figure. Click “Yes” to disconnect the device.

## VI. Successful Connection

After a successful connection, the heart rate sent by peripheral will be displayed on the dashboard screen as shown in Figure.

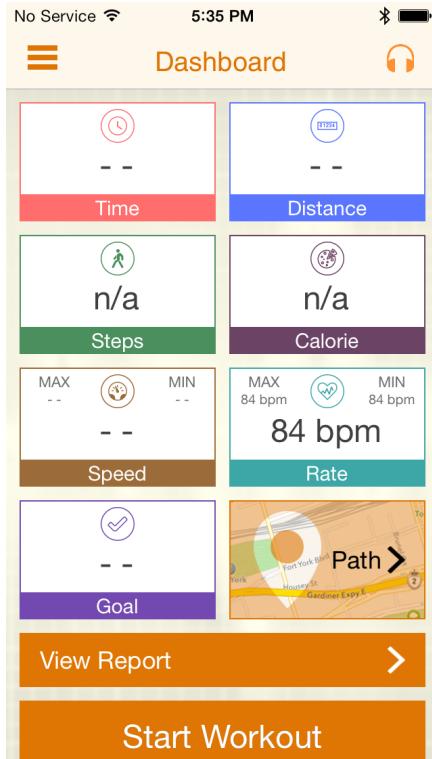


Figure: 7.3.6

## 7.4 Location Services

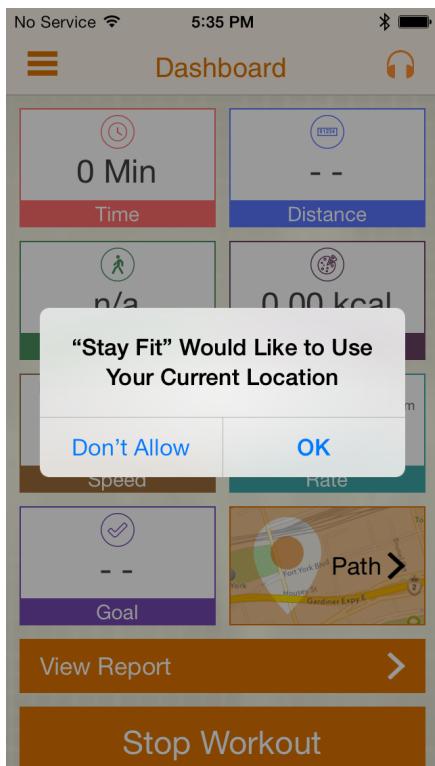


Figure: 7.4.1

### I. Location Service Access Grant

This screen is displayed only for the first time when the user undergoes a workout. Click “OK” to allow ‘Stay Fit’ to access the location services provided by your device.

Note:

The user can also revert this permission in “Settings” application of ios.

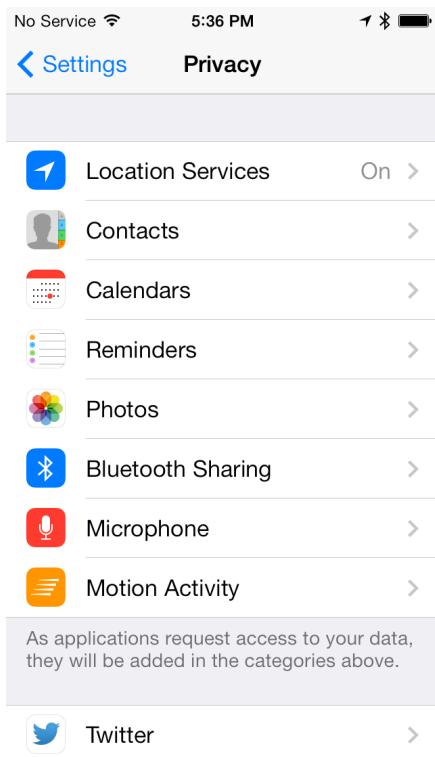
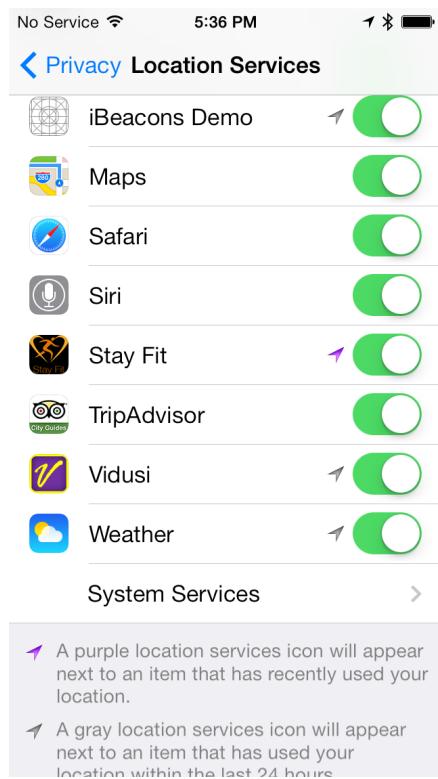


Figure: 7.4.2

### II. Turn on Location Services

Make sure to keep the location services on from the “Settings” app of the device.



### III. Access permission for ‘Stay Fit’

Make sure that under the “Location Services”, the ‘Stay Fit’ app is allowed to use the location data.

Note:

Turning this setting off will effect the proper functioning of the application.

Figure: 7.4.3

## 7.5 Workout

### 7.5.1 Workout Path

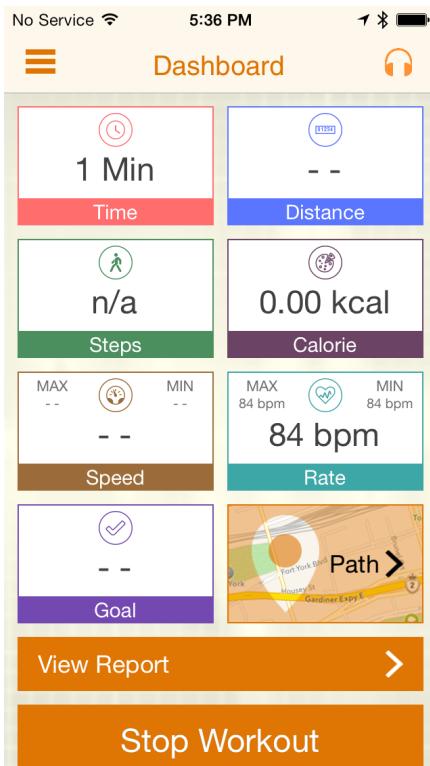


Figure: 7.5.1.1

### I. Dashboard Screen

This screen displays the different attributes of workout as displayed on the screen. These attributes are updated time to time. User can track the progress of the workout from these attributes.

Note:

In the case when no workout goal is set, Click on “Stop Workout” button at the bottom of the screen to end the workout.

By Tapping on the “Path” tab, user can see the current position and the path of the workout as shown in the Figure.

Steps will be available on the device “Iphone 5s” or latter.

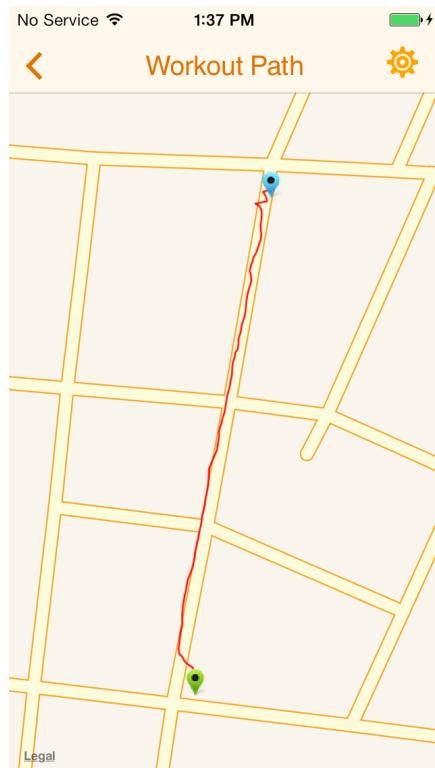


Figure: 7.5.1.2

### II. Workout Path

In these screen the user can track his/her own location by a green annotation on the map and a path by a red colored overlay.

User can also change the type of the map displayed by clicking on the setting button on the top-right corner.



Figure: 7.5.1.3

### III. Map Type

User can change the map type by clicking on one of the 3 options available. The map will be modified accordingly.

Note:

Click in the settings button again to cancel the operation.

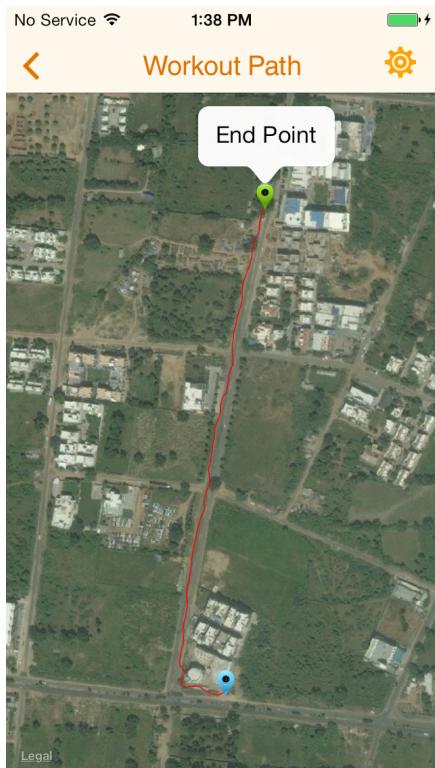


Figure: 7.5.1.4

### IV Hybrid Map

This screen shows a “Hybrid” map. This mode of the map gives better understanding of the current location.

Note:

In hybrid mode of the map acquires more memory on device and requires a better data plan than other modes.

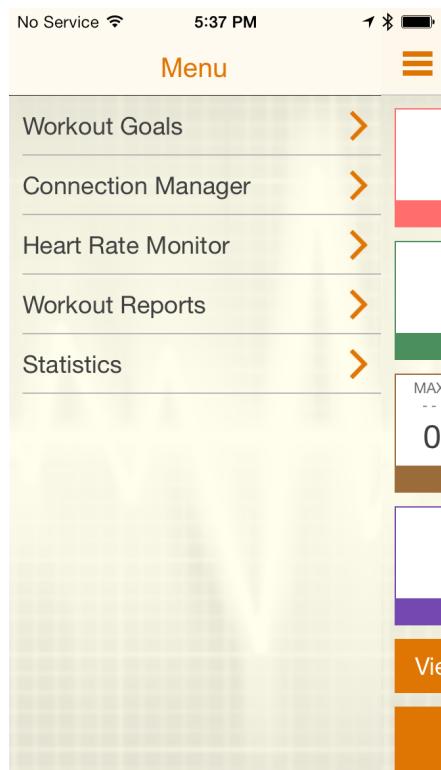


Figure: 7.5.1.5

### V Menu (During Workout)

This screen displays the “Menu” screen while the workout is started. Some of the tabs are not available to avoid malfunctioning during workout.

Note:

As soon as workout is stopped, all the tabs reappear.

### 7.5.2 Workout Reports

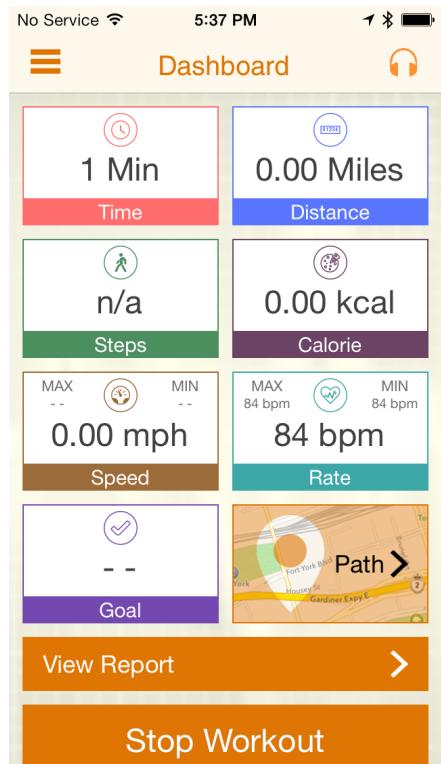


Figure: 7.5.2.1

### I. Navigation to today's Reports

This image displays a tab to be clicked to navigate to a screen where all the report of today's workout is displayed.

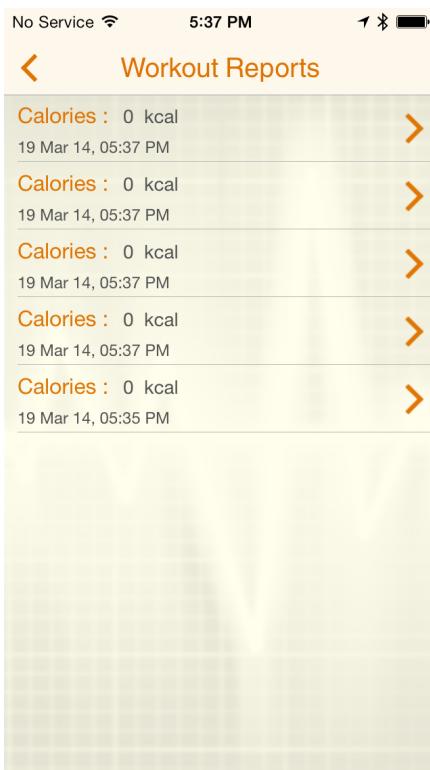


Figure: 7.5.2.2

## II. Today's Workout Reports

This screen displays the summary of all workouts that user have undergone to today. On clicking on the particular report, user will be navigated to more detailed version of the report.

Note:

The date & time displayed in particular tab indicates the start time of the workout.

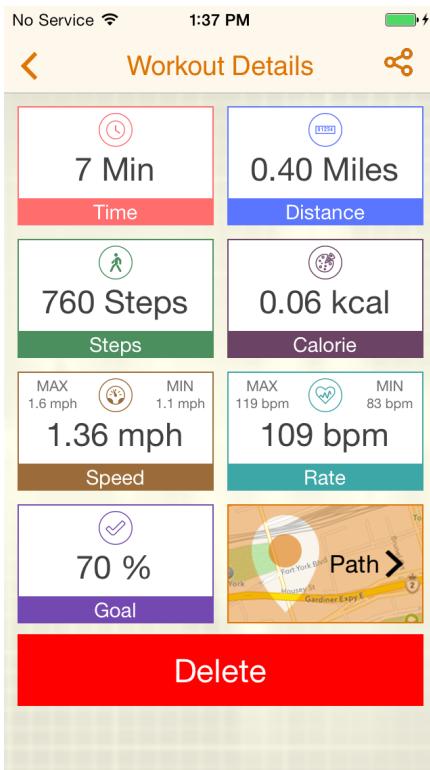


Figure: 7.5.2.3

## III. Workout Details

This screen displays the workout attributes, same as those displayed on “Dashboard”. User can see the workout path by clicking on the “Path” tab.

Note:

User can delete the workout report by clicking on the “Delete” button.

Click on “<” button to navigate back to the previous screen.

Steps will be available on the device “Iphone 5s” or latter.

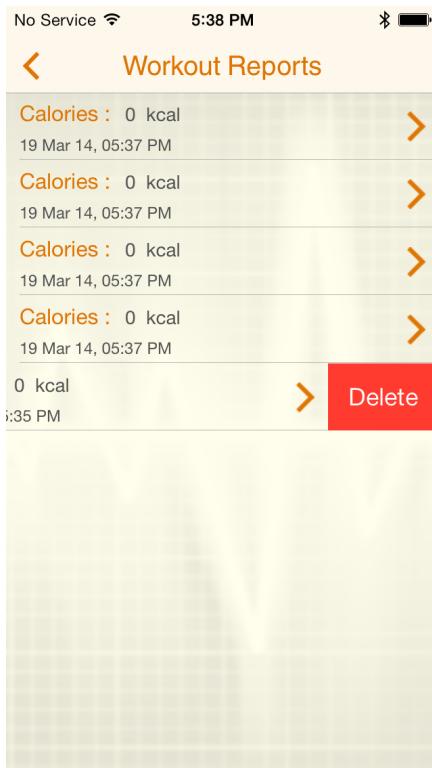


Figure: 7.5.2.4

#### IV. Delete Report

By sliding left on any of the workout summary tab user will see the option to delete it. On tapping “Delete” button the details of the workout will be permanently removed.

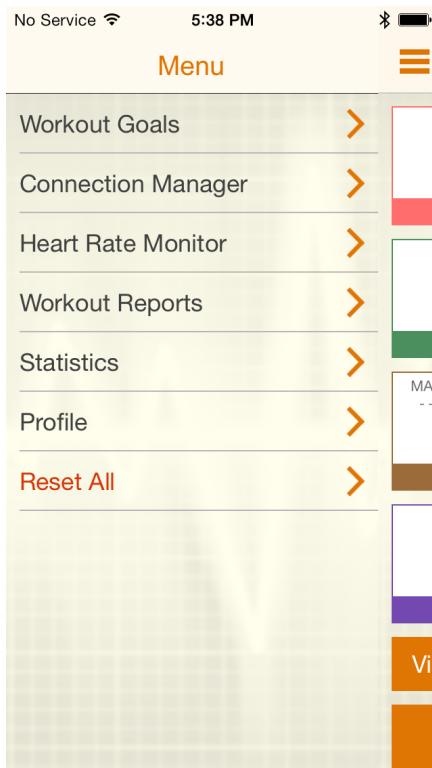


Figure: 7.5.2.5

#### V. Workout Report Navigation

By, tapping on the tab “Workout Reports” in “Menu”, user will be navigated on the screen where details of the all workout till date will be displayed.



Figure: 7.5.2.6

## VI. Workout Reports

Here user will see all workouts he/she has taken till date. User will also be able to select a particular date to review a small set of workouts.

Note:

Click on the top-right button as shown in Figure to filter the workouts.

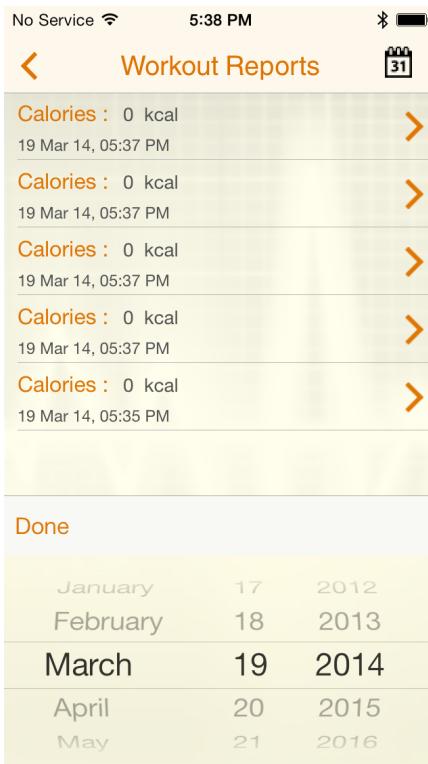


Figure: 7.5.2.7

## VII. Pick a Date for Workout Reports

As seen in the Figure, select the appropriate date to see the workouts, and tap on “Done” button. The workouts will be filtered to the selected date.

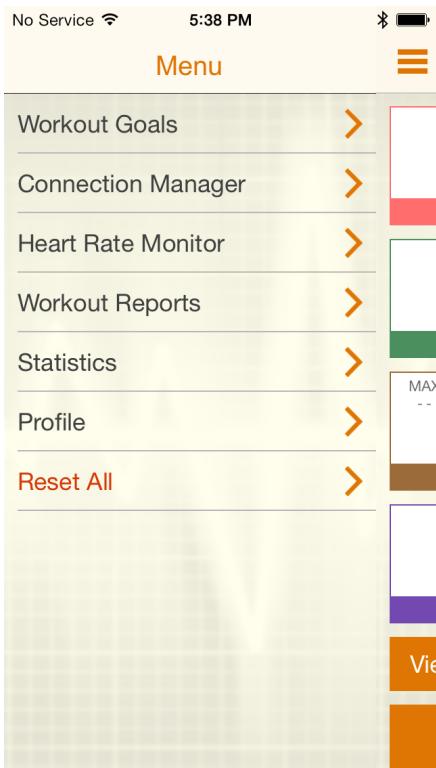


Figure: 7.5.2.8

### VIII. Navigation to Statistics

Click on the “Statistics” tab to view the statistics of user and summary of all workouts in short.



Figure: 7.5.2.9

### IX. Statistics

This screen displays the average, maximum and total of all the workout attributes.

Note:

Steps will be available on the device “Iphone 5s” or later.

Click on “<” button to navigate back to the previous screen.

The Statistics can be reset by “Reset All” tab in “Menu”.

## 7.6 Workout Goal

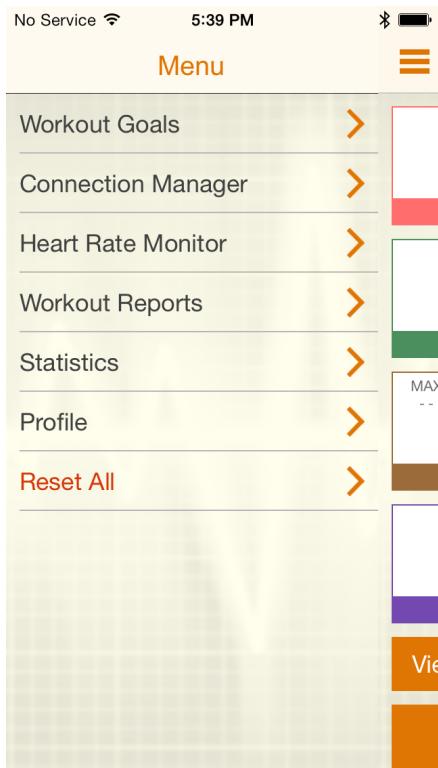


Figure: 7.6.1

### I. Navigation to Workout Goal

Tap on the “Workout Goals” tab to navigate to “Workout Goal” screen as shown in Figure.

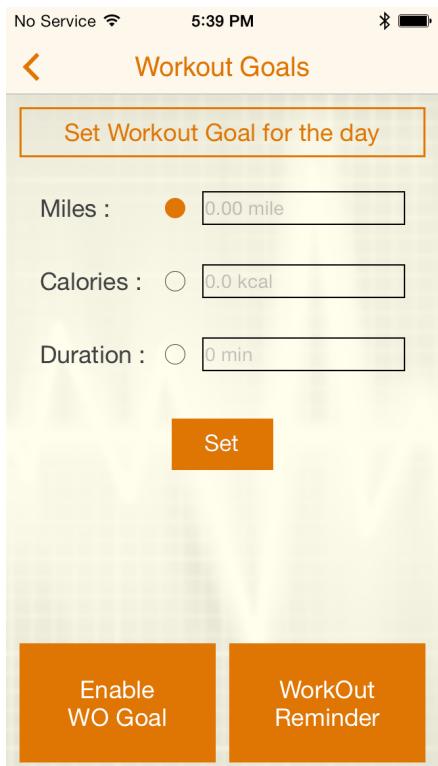


Figure: 7.6.2

### II. Workout Goal

By default workout goal will be disabled when the application is launched first time.

Click on desired attribute of workout enter the value and then tap on “Set” button to set the workout goal.

Note:

After setting the workout goal it needs to be enabled by tapping the “Enable WO Goal” button in bottom-left corner.

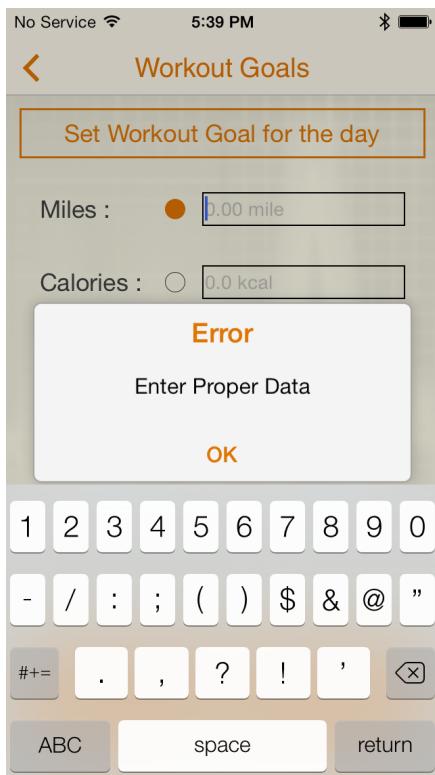


Figure: 7.6.3

### III. Workout Goal Validations

This error is displayed when the field is empty or the data entered into the field is invalid. Enter proper data and try again to proceed.

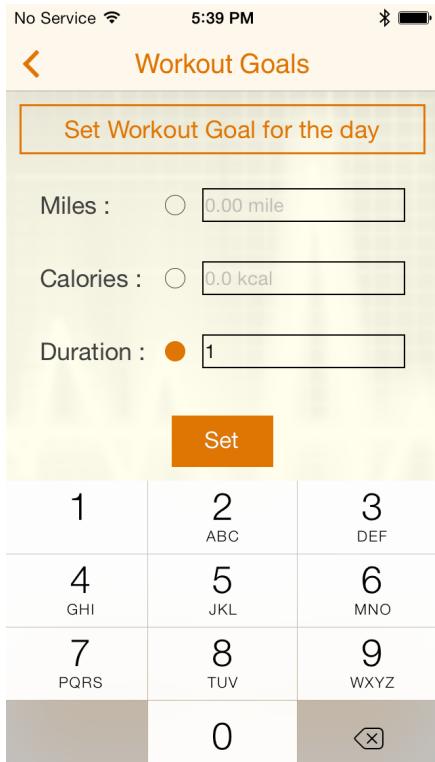


Figure: 7.6.4

### IV. Setting Sample Workout Goal

This shows the sample of setting a workout goal of 1-minute duration.

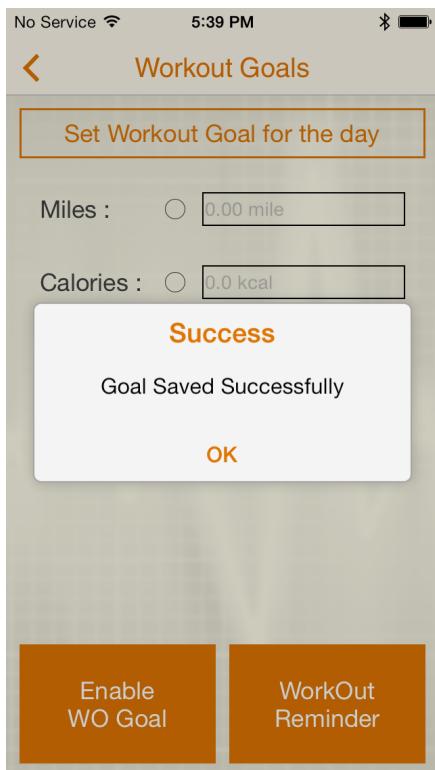


Figure: 7.6.5

## V. Workout Goal Saved

This message is displayed if the goal is successfully saved.

Note:

After setting the workout goal it needs to be enabled by tapping the “Enable WO Goal” button in bottom-left corner.

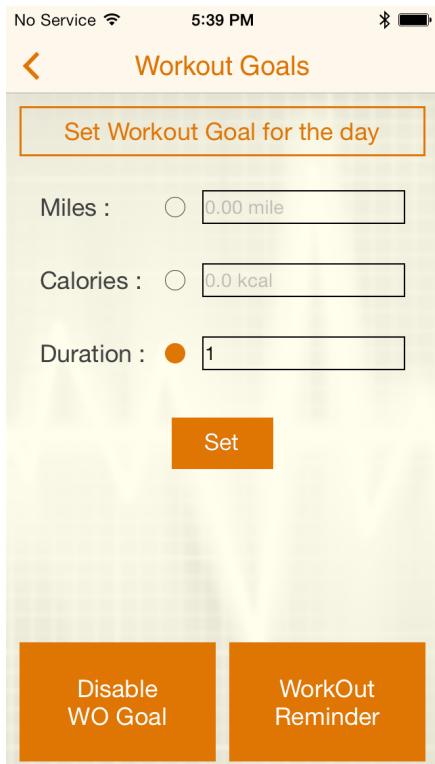


Figure: 7.6.6

## VI. Enable Workout Goal

To enable a workout goal, click on “Enable WO goal” button in bottom-left corner. To revert this, click on the same button again.

Note:

When workout goal is not enabled, the workout process will not be stopped automatically even if a goal is set by user.

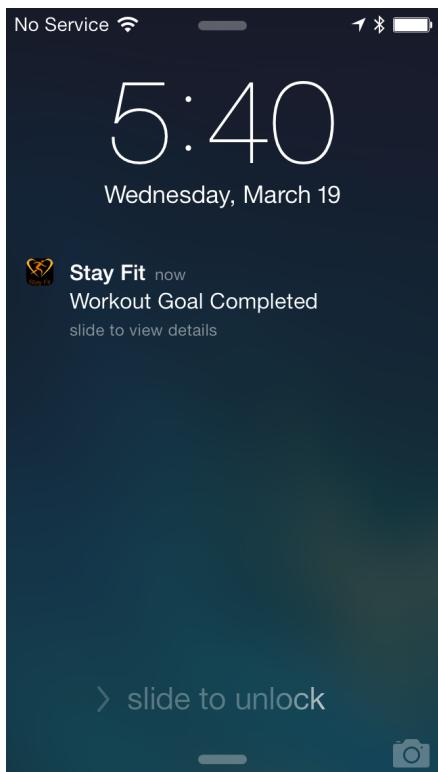


Figure: 7.6.7

## VII. Workout Goal Completion Notification 1

This notification is pushed to device screen when the screen is locked and workout goal is achieved.



Figure: 7.6.8

## VIII. Workout Goal Completion Notification 2

This notification is pushed to device screen when the 'Stay Fit' is not in foreground and workout goal is achieved.

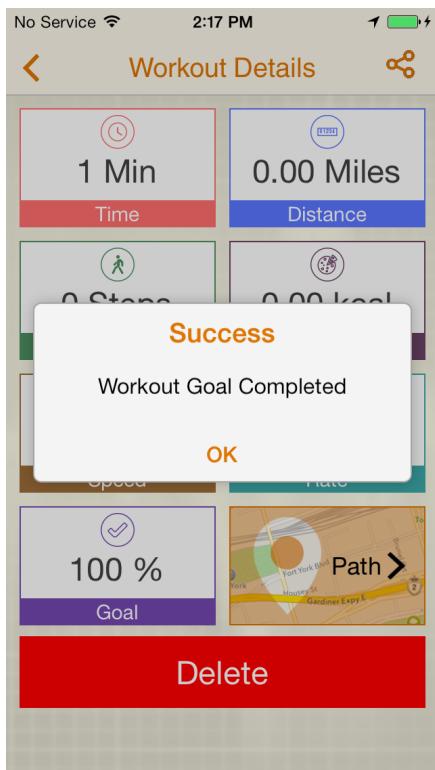


Figure: 7.6.9

## IX. Workout Goal Completion Notification 3

This notification is pushed to device screen when the ‘Stay fit’ is in foreground and workout goal is achieved.

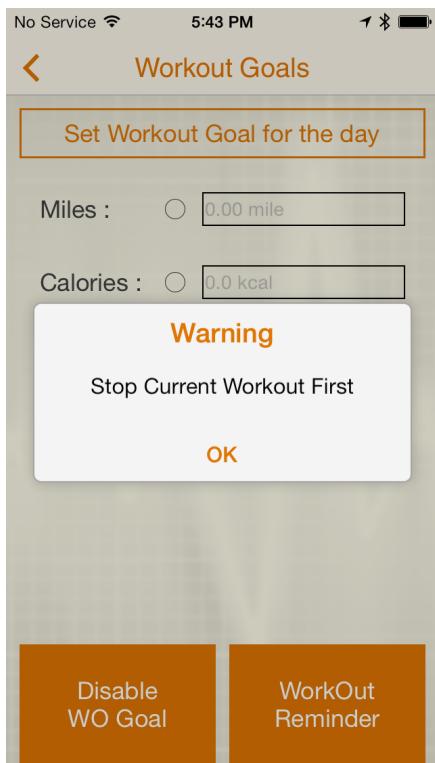


Figure: 7.6.10

## X. Workout Goal Setting Validation

This error is displayed when a workout is in progress and user tries to modify the goal settings.

Note:

Click on “<” button to navigate back to the Dashboard.

## 7.7 Workout Reminder

### 7.7.1 Reminder Permission

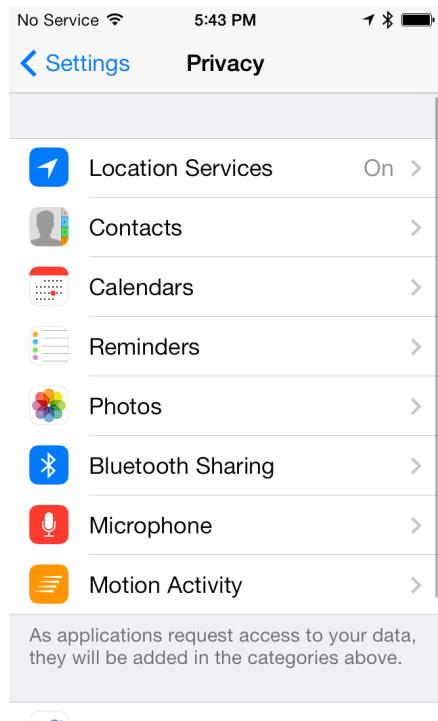


Figure: 7.7.1.1

### I. Reminder Settings

Before accessing the reminder for 'Stay Fit' application it needs to be granted. To do so, go to the "Privacy" option in "Settings" tab.

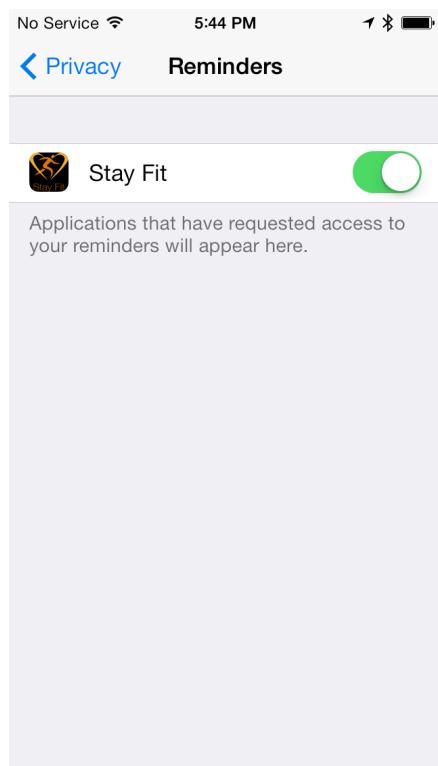


Figure: 7.7.1.2

### II. Reminder Access

Be sure that the access to the "Stay Fit" application is granted as shown in Figure.

### 7.7.2 Add Reminder

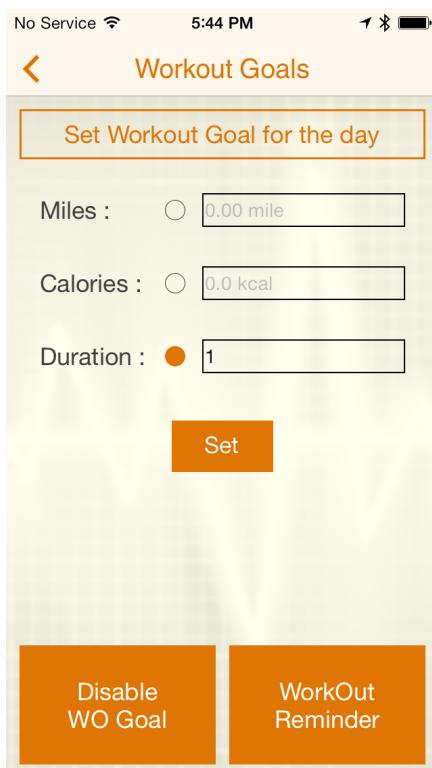


Figure: 7.7.2.1

### I. Navigation to Workout Reminders

To go to the reminder screen tap on “Workout reminder” button in bottom-right corner.



Figure: 7.7.2.2

### II. Workout Reminder List

This screen displays empty reminder list, the case when no reminder is set.

Note:

At this stage tapping on “Edit” or “Delete” button will not have any effect.

Tap on “+” button in top-right corner to add new reminder.

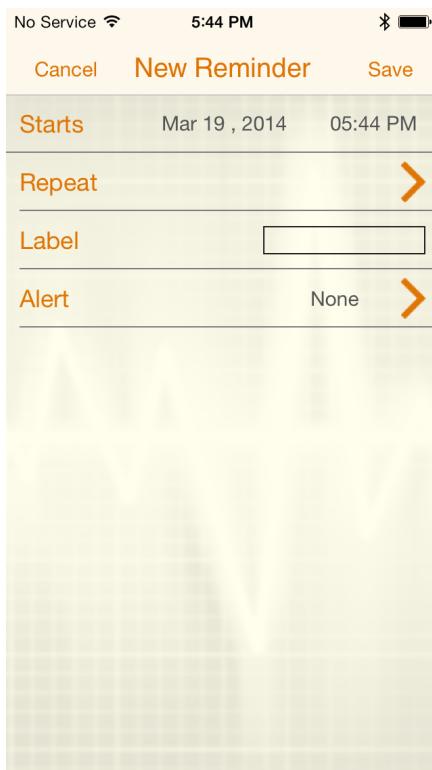


Figure: 7.7.2.3

### III. Add New Reminder

In this screen user can add the details of new reminder and save the same.

Following information about the Reminder is mandatory:

Start date: Date picker

Repeat Pattern: New Screen

Label: Keypad

Alert: New Screen

Note:

All the information is mandatory, missing out any of them will cause a validation error.



Figure: 7.7.2.4

### IV. Reminder Repeat Pattern

Here user can select the days of week on which to repeat the reminder by tapping on the particular day's tab. The orange circle on the tab of the day denotes that the particular day is added.

Note:

At least one of the day in week must be selected. Click on “<” button to go back.

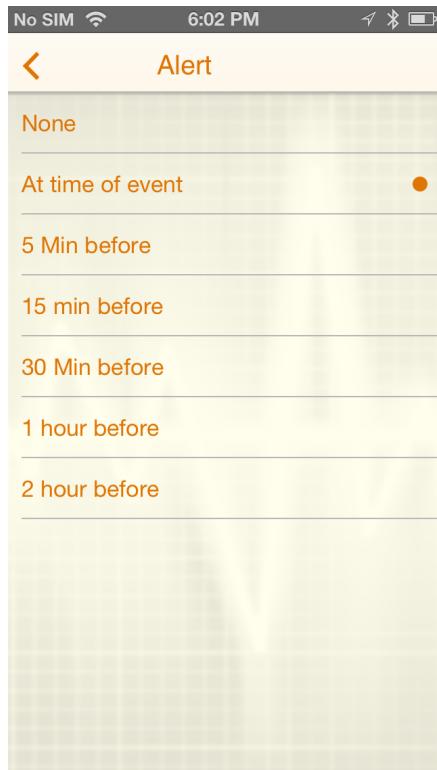


Figure: 7.7.2.5

## V. Reminder Alarm Pattern

Here user will be able to select the time when the device should give an alert to the user to reminder about the purpose of this reminder.

Note:

The alarm is not necessary and by default set to none.

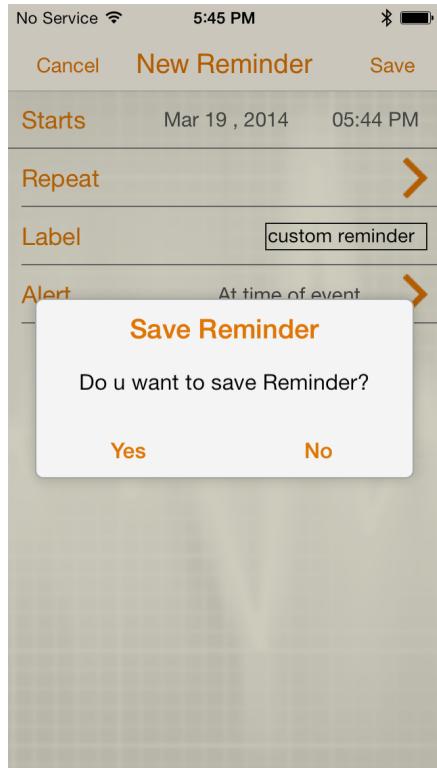


Figure: 7.7.2.6

## VI. Reminder Save Confirmation

On tapping the save button, the user will be asked for the confirmation to save the reminder. Selecting the “Yes” option will navigate the user back to the screen where list of the reminders are displayed.

Note:

On tapping “Cancel” no data about the Reminder is saved.

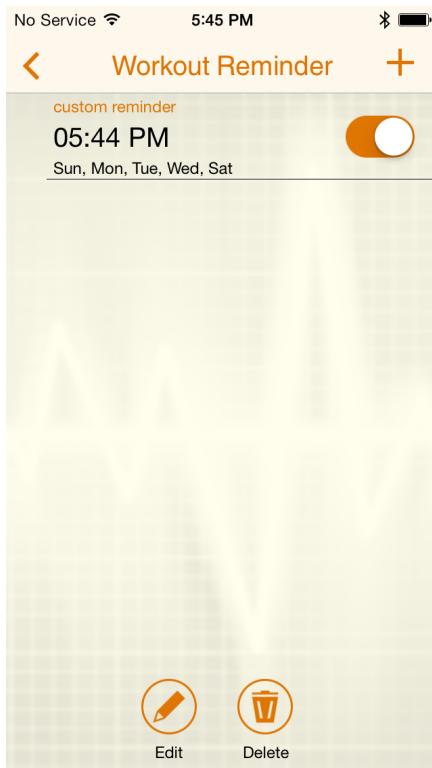


Figure: 7.7.2.7

## VII. List of Reminders

The summary of the reminder is displayed in short in this screen.

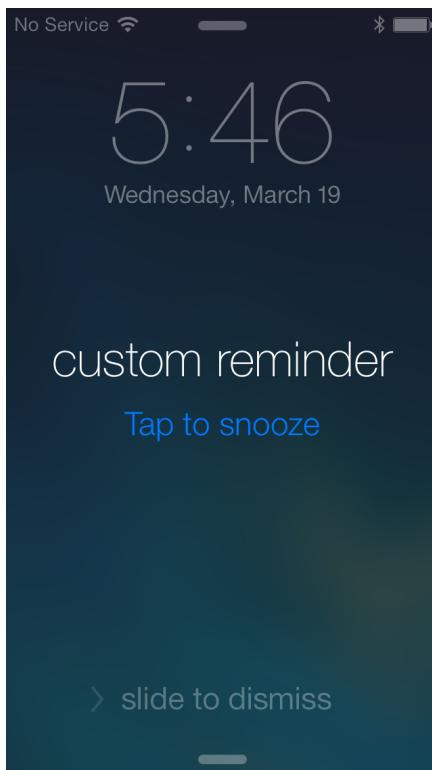


Figure: 7.7.2.8

## VIII. Reminder Alarm Notification

This shows the notification given by the device at the start time of the reminder.

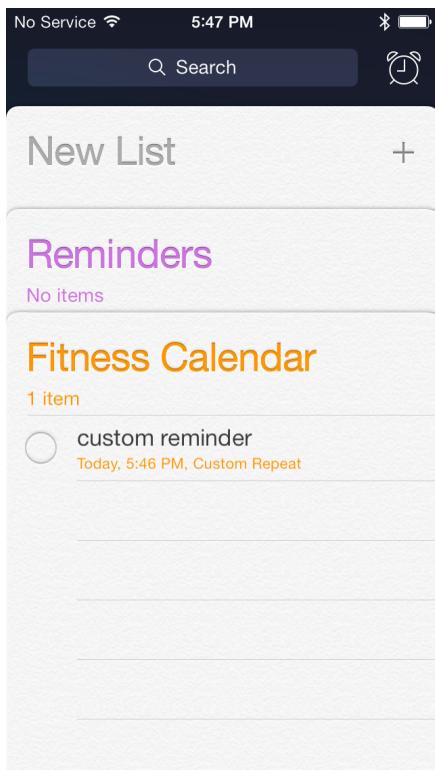


Figure: 7.7.2.9

## IX. Fitness Calendar

This is native reminder application on ios device which displays the number of different calendars. “Stay Fit” creates “Fitness Calendar” automatically.

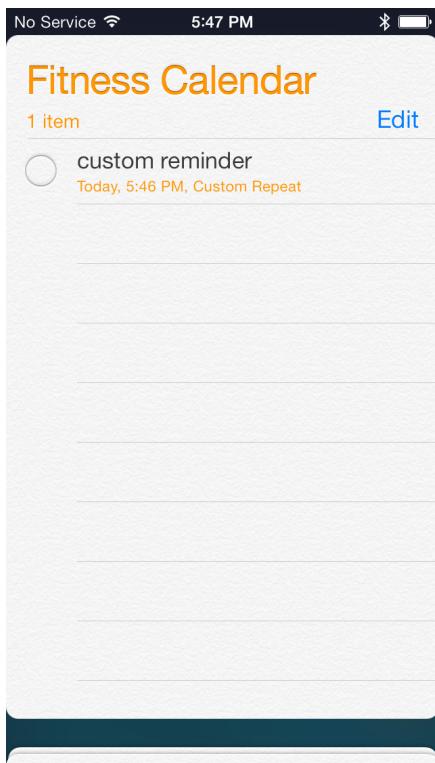
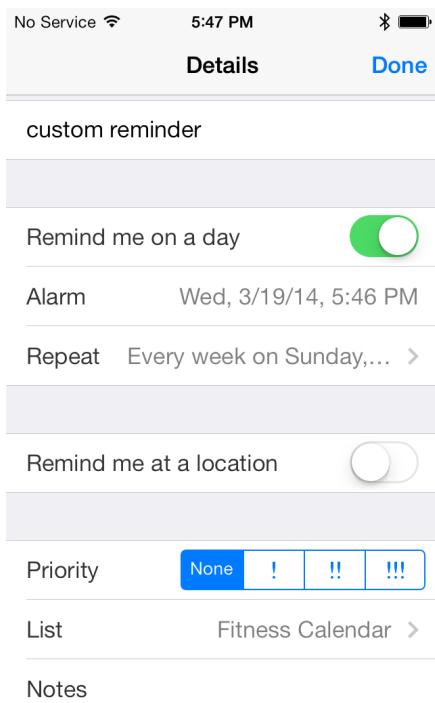


Figure: 7.7.2.10

## X. Reminder in Native App

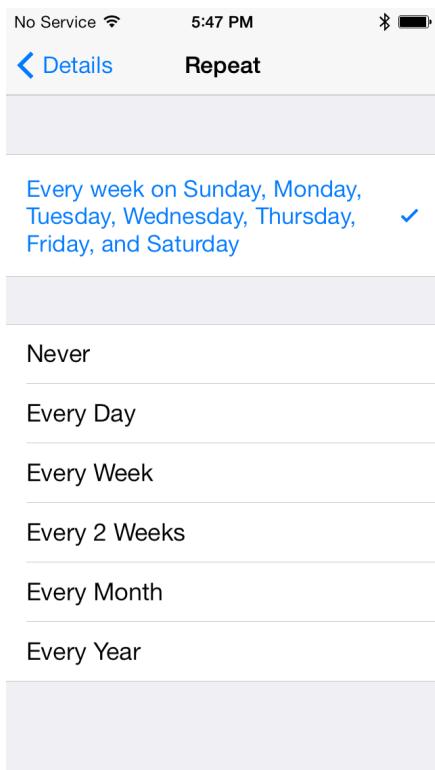
This Figure shows custom reminder in native app created by ‘Stay fit’.



## XI. Reminder Details

This Figure shows the details of the custom reminder.

Figure: 7.7.2.11



## XII. Repeat Pattern in Native app

This Figure shows a custom repeat pattern of custom reminder.

Figure: 7.7.2.12

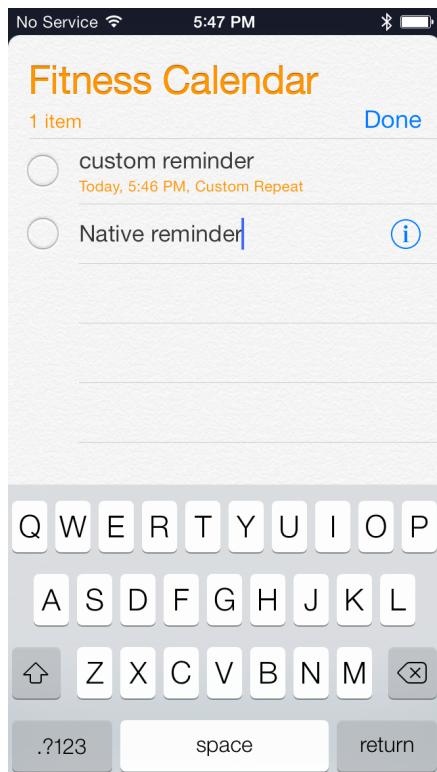


Figure: 7.7.2.13

### XIII. Add New Reminder in Native App

Here, by tapping “+” button on top-right corner, user can add new reminder in “Fitness Calendar”.

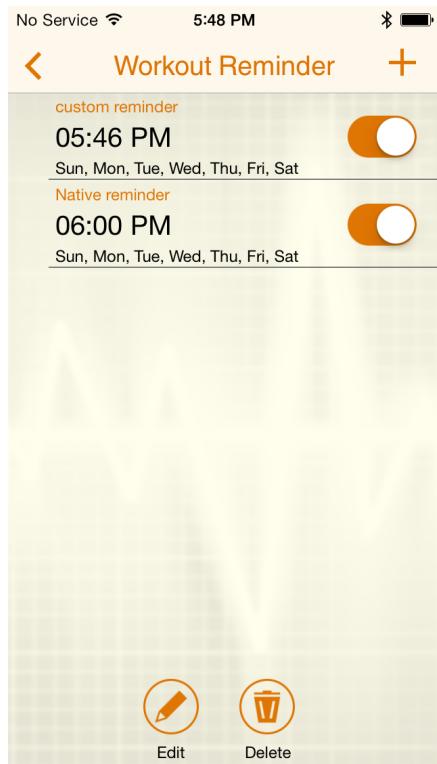


Figure: 7.7.2.14

### IXV. Reminder Fetch

The reminder created in native app is automatically fetched in ‘Stay Fit’.

### 7.7.3 Edit/Delete Reminder

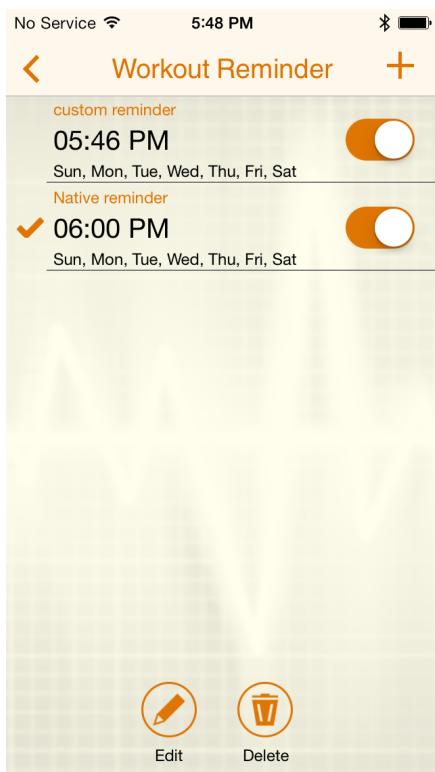


Figure: 7.7.3.1

### I. Select Reminder

To edit a reminder, just tap on it and click on “Edit” button at bottom-left.

Note:

A checkmark symbol is displayed against the selected reminder.



Figure: 7.7.3.2

### II. Edit Reminder Screen

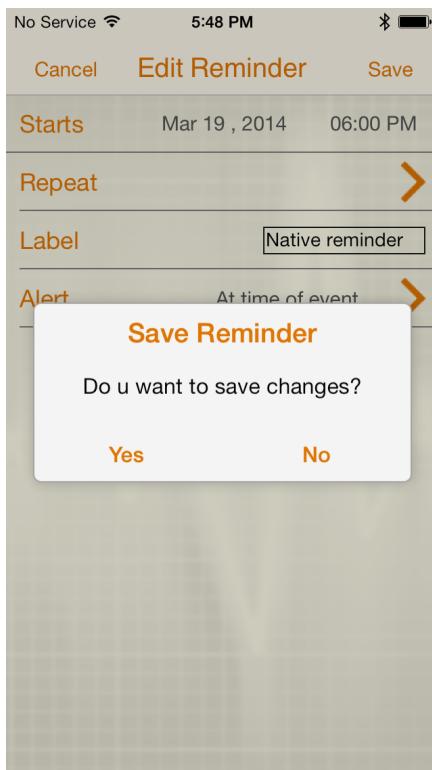
This screen is same as the adding new reminder details. The user is required to fill the details appropriately.



### III. Change Repeat Pattern

Change the repeat pattern of the reminder by tapping out the days on which the reminder is not required.

Figure: 7.7.3.3



### IV. Save Changes Confirmation

On tapping the save button, the user will be asked for the confirmation to save the reminder. Selecting the “Yes” option will navigate the user back to the screen where list of the reminders are displayed.

Note:

On tapping “Cancel” no data about the Reminder is saved.

Figure: 7.7.3.4

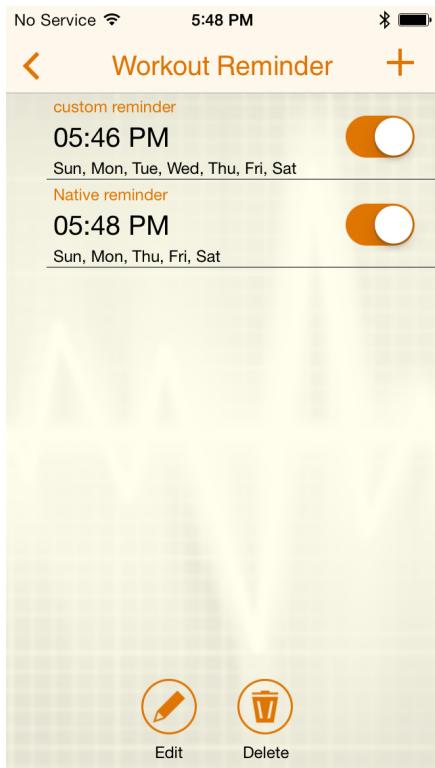


Figure: 7.7.3.5

## V. Changes Updated

Here, a change in a previously saved reminder can be seen in the summary.

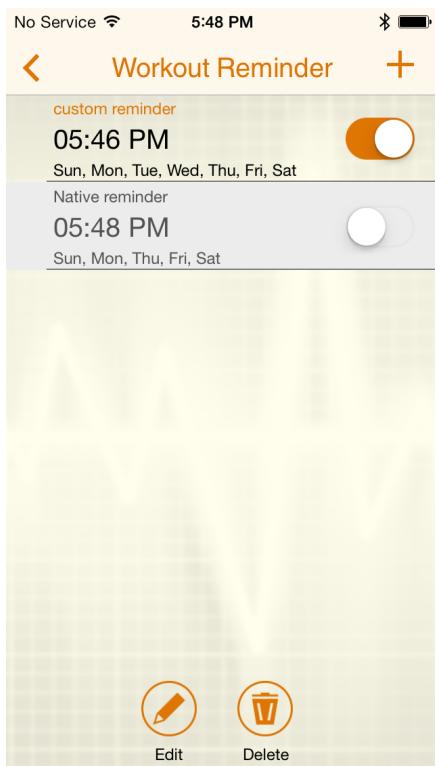


Figure: 7.7.3.6

## VI. Enable/Disable Reminder

To turn the reminder off (to remove an alarm), just turn off the switch and the alarm from the reminder will be removed.

Note:

Reminder still will have it's effect and would be accessible from the native app.

To revert the changes, change mode of the switch to “on” state again.

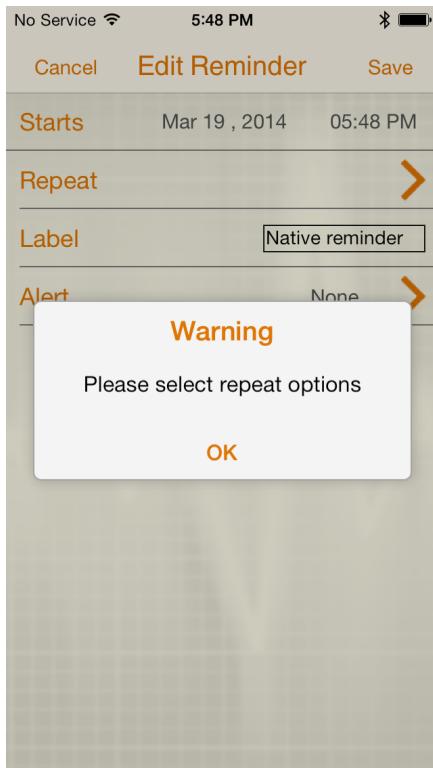


Figure: 7.7.3.7

## VII. Repeat Pattern Missed Out

This error message is displayed when user misses out to set the repeat pattern for a reminder.

Select the repeat pattern and try saving it again.

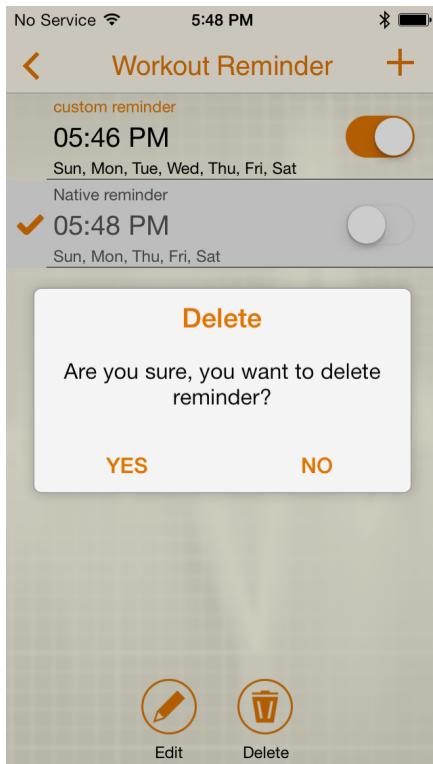


Figure: 7.7.3.8

## VIII. Delete Reminder Confirmation

When user selects a reminder and clicks on “Delete” button in bottom-right corner, this conformation message allows user to avoid accidentally deleting a reminder.

By selecting “Yes”, the reminder will be permanently deleted.



Figure: 7.7.3.9

## IX. Reminder Deleted

Here the reminder is deleted from the calendar and user will not be able to receive any notification on the start date of the reminder.

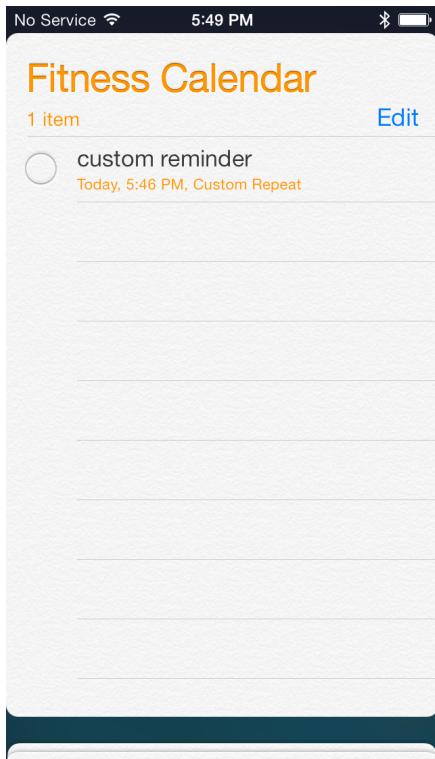


Figure: 7.7.3.10

## X. Reminder Deleted in Native App

The reminder is deleted from the native app also and all the changes made to a reminder in 'Stay Fit' are also propagated to the native app as well.

## 7.8 Heart Rate Monitoring

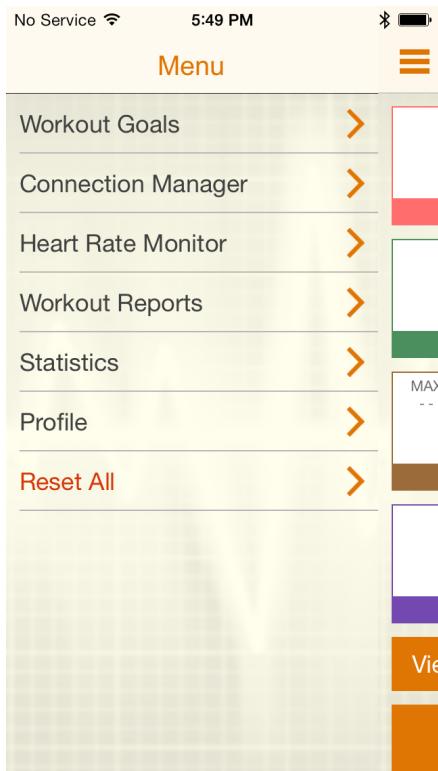


Figure: 7.8.1

### I. Navigation to Heart Rate Monitor

Tap on “Heart Rate Monitor” tab in “Menu” to navigate to HRM.

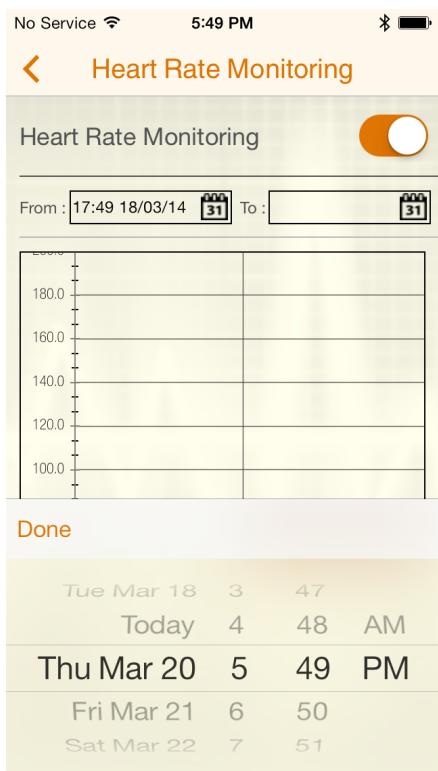


Figure: 7.8.2

### II. Filling Filtering Details

Fill the duration during which the graph is required to be plotted on the values of heart rate of user.

Note:

Here user can turn off the continuous monitoring of Heart-Rate by switching the switch to “Off” state. This will have the same effect as the switch present in the user profile.

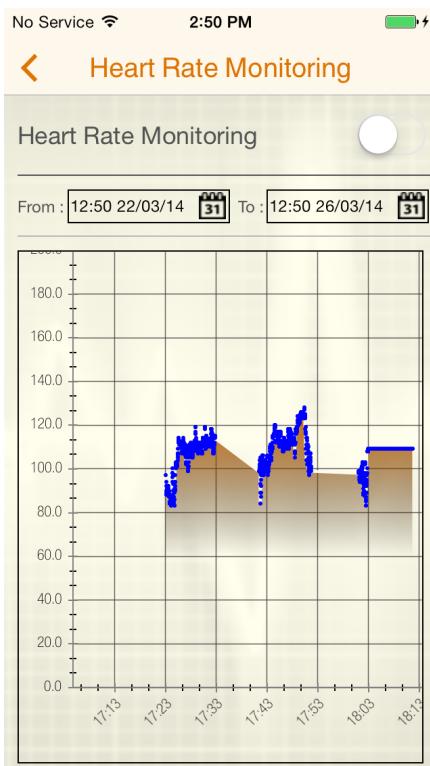


Figure: 7.8.3

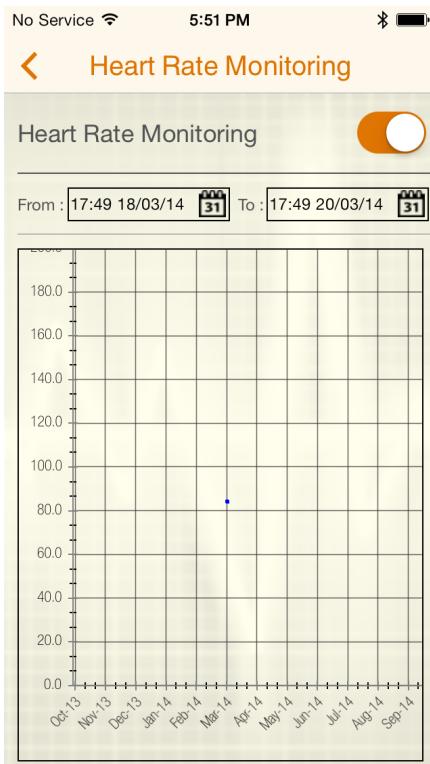


Figure: 7.8.4

### III. HR Graph (hh:mm)

Here the graph is plotted according to the availability of the data and the unit as well as zoom level is detected to suit the duration selected by the user.

### IV. HR Graph (MMM-YY)

This screen shows the maximum zoomed out version of the graph.

Note:

Pinch gesture is used to zoom in or zoom out the graph. The units of X axis are modified according to zoom level.

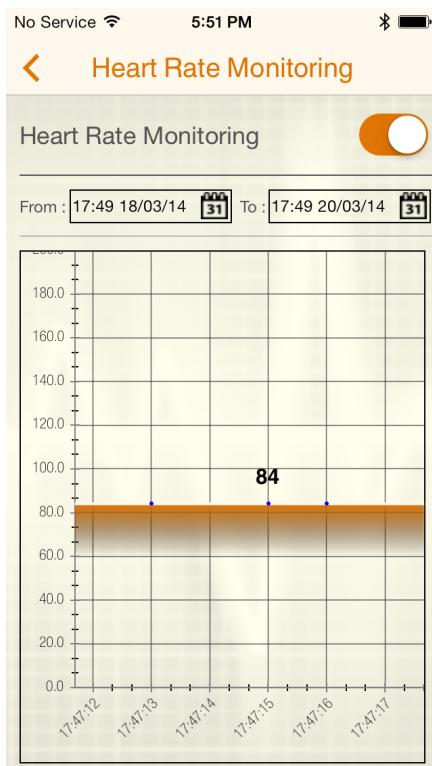


Figure: 7.8.5

## V. HR Graph (hh:mm:ss)

This screen shows the maximum zoomed in version of the graph.

Note:

Pinch gesture is used to zoom in or zoom out the graph. The units of X axis are modified according to zoom level.

On tapping any of the blue dot, the actual value of the heart rate at that point of time will be displayed as shown in Figure.

## 7.9 Music Player

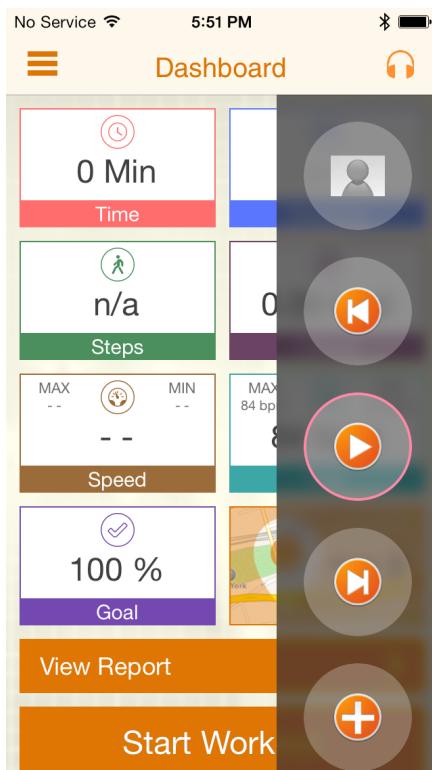


Figure: 7.9.1

### I. Showing Music Player Controls

Here, in Dashboard screen on tapping a music icon a music tray is displayed. It can be hidden again by tapping the same button again or by tapping any area outside the tray.

On tapping “+” button as shown in Figure, user will be redirected to the music library to add the songs to the custom playlist.

Note:

On tapping “Play” button, no action will be taken if the music library is empty or the playlist is not created.

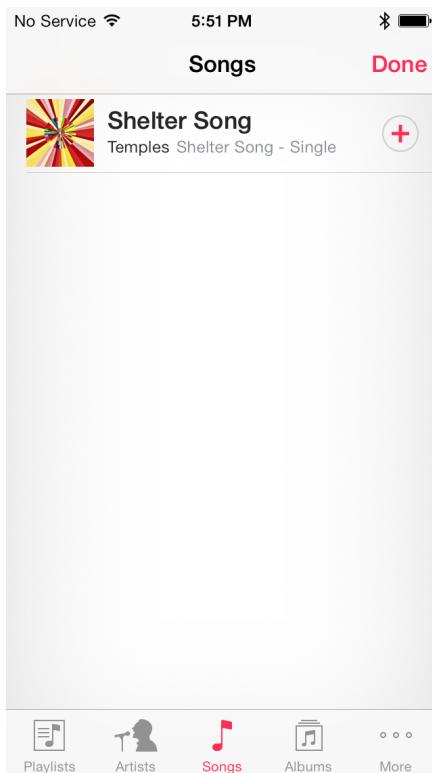


Figure: 7.9.2

### II. Add Songs to Playlist

This screen is from the native music application from where the songs will be added to custom playlist. Tap on “Done” button to complete the operation.

Note:

The music will start playing when the user returns to the ‘Stay Fit’ app, if at least one song is added to the library.



Figure: 7.9.3

### III. Music being Played

Here the music is being played currently and other controls in the tray are also active now.

Play/Pause  
Next track  
Previous Track  
Track Artwork

Note:

The Artwork of the track is for display purpose only and hence it will not result into any effect when tapped.

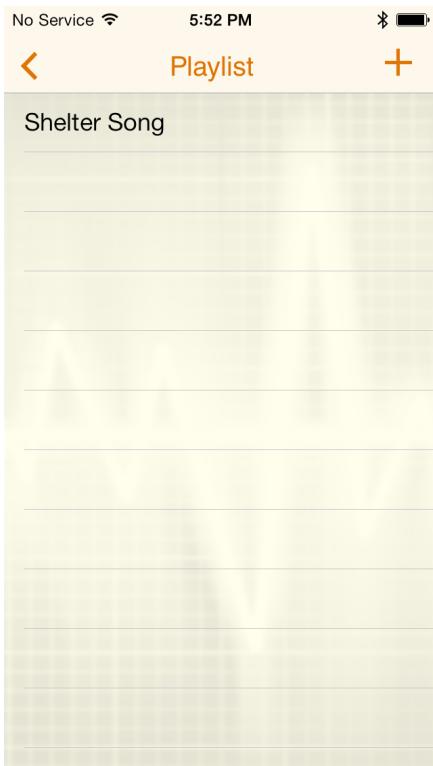


Figure: 7.9.4

### IV. Playlist

This screen shows the tracks, those added by the user from the library. On tapping “+” button, user will be displayed a library and can add few more songs.

Note:

To play a particular song, tap on the tab with the name of the track on it.

## 7.10 Reset All Data

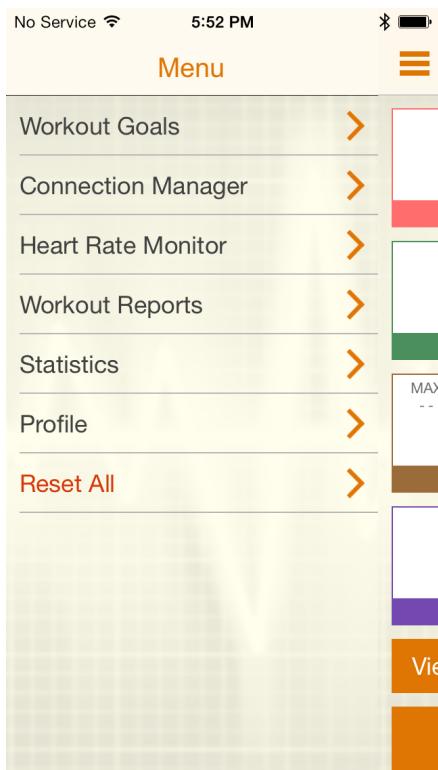


Figure: 7.10.1

### I. Reset All Data

Tapping on the tab “Reset All” in the “Menu” will reset the application.

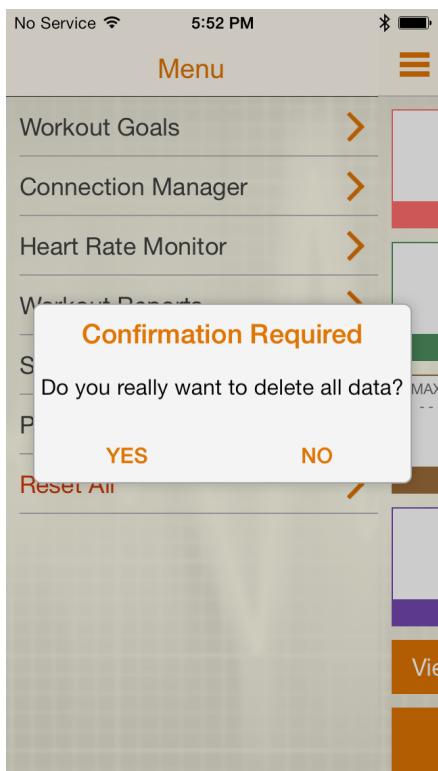


Figure: 7.10.2

### II. Confirmation of Resetting Data

Before resetting the app, a confirmation will be required. On selecting “Yes”, all the user data will be deleted and application will be reset to state when it was first installed.

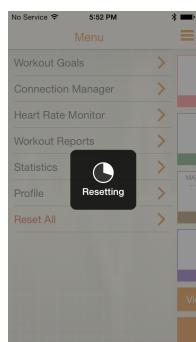


Figure: 7.10.3

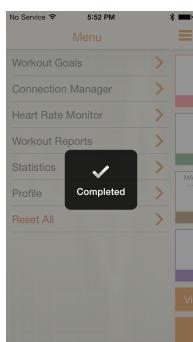


Figure: 7.10.4

### III. Progress Of Resetting Data

The process of deleting and resetting the app requires a few seconds, and after which the application is navigated to the screen “User Profile”, and user is prompted to add all the details again.

Note:

This is required when a new user starts using the application.

The “Fitness Calendar” is also deleted along with all the reminders.

## 7.11 Social Network Sharing

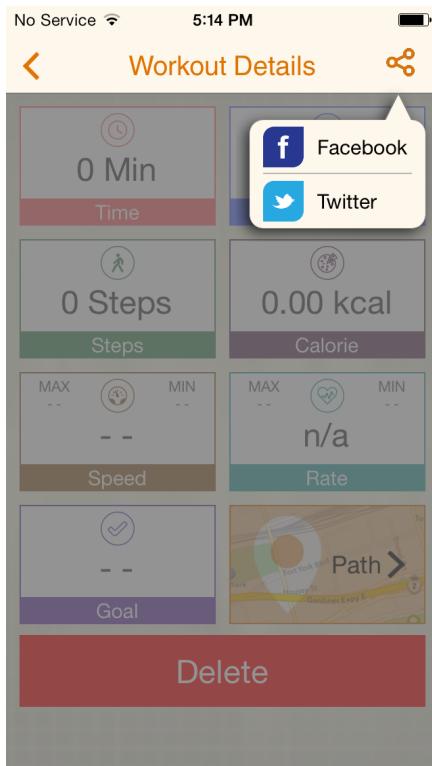


Figure: 7.11.1

### I. Sharing Options

On Clicking the share button in top-left corner, the user will be allowed to select the option of either to share on Facebook or Twitter.



Figure: 7.11.2

### II. Account Configuration

The user is required to add an account of facebook or twitter to the settings app of iphone.

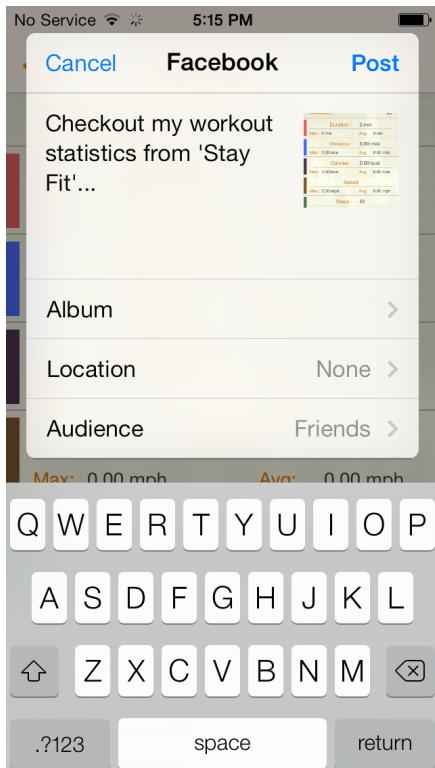


Figure: 7.11.3

### III. Facebook Dialog (Statistics)

The dialogue shown in the Figure is the control that allows user to share the data on Facebook.

Click on “Post” to post the status update or click “Cancel” to go back.

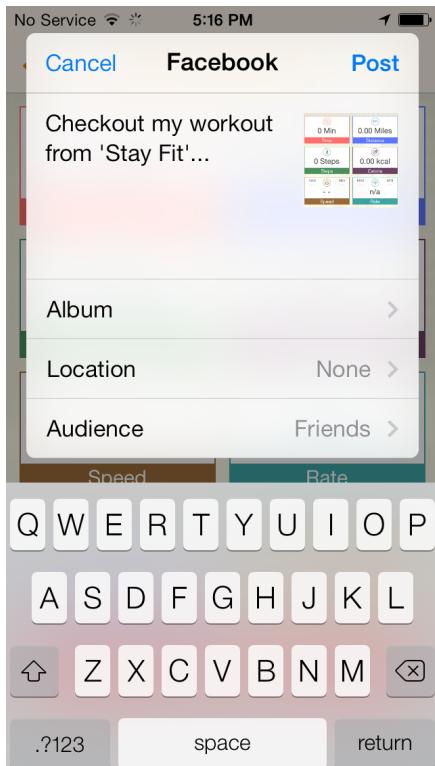
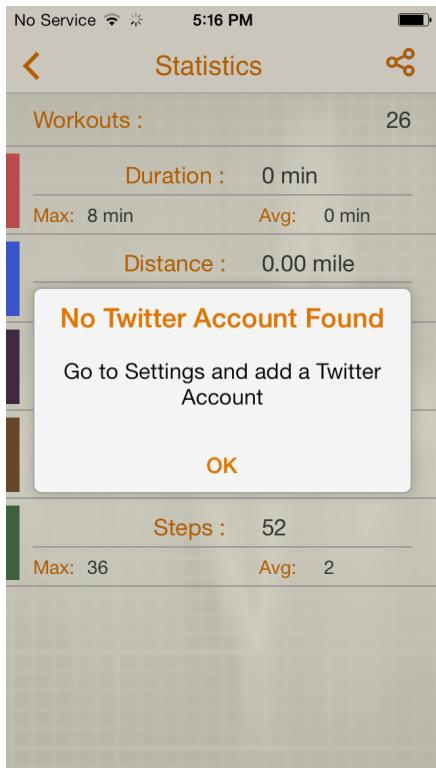


Figure: 7.11.4

### IV. Facebook Dialog (Workout)

The dialogue shown in the Figure is the control that allows user to share the data on Facebook.

Click on “Post” to post the status update or click “Cancel” to go back.



## V. Account Configuration Missing

This error is shown when the user tries to share the information without configuring any account.

Figure: 7.11.5

## **Chapter 8**

### Conclusions & Future extensions

## 8.1 Limitations

‘Stay Fit’ application runs in only ios devices and that also iphone only. The support for the other platforms is not still provided. It also requires the BLE to function properly. ‘Stay Fit’ runs in only portrait orientation of the device, it does not automatically adjust the UI according to the change in device orientation. So user will see the uniform user interface throughout the application.

The application allows only one user to use the functionality at a time if other user wants to use the application then it must be reset before he/she can do so. When application is reset, all the data previously saved for the user are deleted. Also the data for the user is saved locally on a device and provides no backup. If user changes his/her device then all the data those were saved on old device cannot be restored to new one. Also if the user uses multiple devices the synchronization support between two different devices is not provided. Also there is no backup for this locally saved data.

The voice assistance support is provided periodically during a workout only, so user is not able to customize the duration after which a voice feedback should be provided. All the parameters of voice assistance are predefined and cannot be changed. Though user have option to turn off or on the voice support.

The user will not be able to delete the specific heart rate data of certain duration, though the workout data can be deleted which also deleted all the attributes of workout along with heart rate data.

If user makes any changes to the workout reminder in the native reminder application of ios then those changes are reflected to the application but the application is not able to prevent the changes from the native app.

The application requires the user to have a preconfigured facebook or twitter account before sharing the workout attributes it does not provide the mechanism to authenticate the user to the appropriate network account.

## 8.2 Discussion

### 8.2.1 Self Analysis of Project Viabilities

We started the project first by understanding coding for Objective-C. We already know primary language C and object oriented languages C++ and JAVA , so things were a little easy, but there was a lot to learn. After getting some basic information about it, we started learning about BLE technology and BLE enabled devices. Then the in depth study of “180D” service of BLE was undertaken.

We developed the whole project under the guidance of senior iPhone developer working at Infostretch Solutions. He is proficient in mobile development and he helped us to solve the major portions of the doubts that we had while developing the application. Implementation of ‘Stay Fit’ Mobile application took the major portion of time as we have used MVC(Model-View-Controller) architecture

and the most difficult portion of application was to write a custom library that would help any BLE peripheral to connect to the BLE enabled device and also provide sufficient durability to save bulk data. Once that was complete, we had confidence that the project would reach a successful conclusion.

‘Stay Fit’ uses a peripheral device, which has implemented the HRM (Heart Rate Monitor) profile of Bluetooth smart protocol. The application gets user’s heart rate in bps unit from the device and also records the location details and allows users to track his/her fitness, all in one app. In addition to that, the application keep record of user’s daily activity and also shows a statistics of all workouts that user has undergone to. The application also uses GPS to accurately measure the distance covered and speed of user during workout. Over all, ‘Stay Fit’ helps user to stay fit.

The Graphical user interface of the application is so nice and simple that any novice user can handle the application very easily. Also, if user faces some difficulties in handling application, we have provided help module having snapshots to show handling of application.

### **8.2.2 Problems Encountered and Possible Solutions**

Initially the problems we faced were in understanding the Objective-C, iOS platform and Xcode Development tool, their features and various controls used for development of iPhone application, as it was completely new for us. These were solved by the online documentation provided by Apple. Also, our project guide has given us many demos regarding development of it.

In the project, the most difficulty was experienced when plotting the heart rate data on a graph. That required in depth understanding of the core plot and the open GL framework, which is very different from the rest of the modules of the application. Various tutorials and example code helped a lot in saving time and providing custom and best possible solution to plot HR data on graph that automatically zooms in and out and also changes axis labels accordingly.

Main difficulties that are commonly faced by all iOS Developer is that regarding Memory management. Because, there is no any thing named garbage collection in iPhone. There are commonly so many memory leaks are observed in many iPhone application.

The vast difficulty was to provide support for user interface of iOS 6 and 7 both. There was a vast change in the UI when iOS 7 was introduced. It was very difficult to design screens, which look exactly similar in both iOS.

One of the new feature introduced in ios was auto-layout. Before this the components of UI were designed using auto-resizing masks. All the working employees of the company were using that only, thus there was not much experience on the new technology. Infostretch organized a seminar on “Auto-Layout” to make the employees familiar with new technology. After various sample applications and apple guidelines, the project UI was successfully developed using auto layout constraints.

### 8.2.3 Summary of Work Done

We have been working on iOS Platform since December and we have successfully completed this project as per requirement. We have developed our project using OS X 10.8.2 (Mountain Lion). We have used MVC model for developing the project. For providing simple and nice view to our application, we have used nib interface.

To be a bit more precise about the learning during the project work we would like to mention the technical knowledge gained about MVC model, UI design etc. The greatest learning curve during these months was learning about the work culture in industries, the interpersonal skills gained through the teamwork and efficient work patterns to resist against stress and work pressure yet achieve the targets given.

## 8.3 Conclusion

Obesity is the most prevalent, fatal, chronic, relapsing disorder of the 21st century. Obesity is a leading cause of world's mortality, morbidity, disability, healthcare utilization and healthcare costs. It is likely that the increase in obesity will strain our healthcare system with millions of additional cases of diabetes, heart disease and disability. Significantly, excess adiposity or obesity causes insulin secretion, which can cause insulin resistance that leads to type-2 diabetes.

Regular physical activity will help you to avoid these things. Physical activity is essential to prevent and reduce risks of many diseases and improve physical and mental health. It can even help you live longer—research from the American Journal of Preventative Medicine indicates that regular exercise can add up to five years to your life.

Since regularity in physical activity is very important, the development of the fitness application that can run in your mobile device is necessary.

This mobile application can be used to track different parameters during workout. This app will be able to connect to BLE enabled Heart Rate Monitor. This monitor will continuously send user's Heart Rate data to application. This application will keep track of Heart Rate data along with user's Speed (Min, Max & Avg.), Energy Expended (Calories burned), Duration, Distance, Step Count, Path on Map. User will be able to set his/her workout goals through this application. Application will notify user on completion of any goal. User will be able to view/share his/her daily/weekly/monthly workout statistic from app. This app will remind user for Workout if he/she has set any reminder. User will be able to Play music during workout.

## **8.4 Future Extensions**

### **Cloud Synchronization**

As per the current limitation of the application, the locally saved data is not backed-up anywhere. Thus on changing of the device, the data is lost and user has to start it all again. Thus this application can be extended to provide the support for synchronizing data to cloud, hence user cannot even get his/her data back on new device, he/she will also be able to use multiple devices at same time and all the data would be saved and retrieved from iCloud.

### **Plotting ECG**

Currently user is able to see only the heart rate data on graph. The BLE peripheral also sends the RR interval values via Bluetooth which can be used in plotting ECG of the user. That ECG can be very useful and it has large applications of its own, in the area of medical science.

### **Social Network Synchronization**

Using ‘Stay Fit’, user can share the attributes of his/her workout on facebook and twitter. But user is not able to see the progress of his/her friends that may be shared through the same application on the social networking sites. The application can be extended to provide support for user to add the friends from his/her social network account and see the details of their workout and also compete with them to achieve better fitness results.

### **Support for other BLE Device Connection**

Currently ‘Stay Fit’ scans and connects to BLE devices with service uuid ‘180D’. It can be extended to allow user to connect other BLE devices such as heart Rate Monitors, which have added functionality.

## **Bibliography**

### **Books**

- Beginning Iphone SDK Programming With Objective-C
- Creating Iphone Apps With Cocoa Touch The Mini Missing Manual
- Developing Enterprise Ios Applications
- Learning Iphone Programming
- Programming in Objective-c - 4<sup>th</sup> Edition

### **Websites**

- <https://developer.apple.com/library/ios/>
- <https://www.google.com>
- <http://www.wikipedia.com>
- <http://www.raywenderlich.com/tutorials>
- <http://www.icodeblog.com>
- <http://www.techotopia.com/index.php>
- <http://iphonedevsdk.com>
- <http://www.stackoverflow.com>
- <http://www.code4app.net>
- <http://www.github.com>

**Project Guide Details**

Name of External Guide	Mr. Kuldeepsinh Kumpavat
Designation	Senior Software Engineer
Contact Number of Guide	9687964123
Email Id of External Guide	kuldeepsinh.kumpavat@infostretch.com
Name of Company	InfoStretch Solutions Pvt. Ltd.
Contact Number of Company	Off - +91-79-40266266
Address of Company	Infostretch Solutions Pvt. Ltd, Neptune Corporate House, 101-105 Block A, B/H Rajpath Club, S G Highway, Ahmedabad - 380054