CMIS 495 7980

Professor Howard

Spring 2017

Group 4

Ryan Carnes

Anabelle Baez Rodriguez

Gregory Fraser

Aaron Smith

26 March 2017

Version 1.0

**FitTrac System**

**Software Project Plan**

# Table of Contents

[Table of Contents ii](#_Toc478325991)

[Preface 3](#_Toc478325992)

[Overview 3](#_Toc478325993)

[1. Project Summary 3](#_Toc478325994)

[1.1 Purpose, scope, and objectives 3](#_Toc478325995)

[1.1.1 Required Features 4](#_Toc478325996)

[1.2 Assumptions and constraints 4](#_Toc478325997)

[1.3 Project deliverables 4](#_Toc478325998)

[2. Project organization 6](#_Toc478325999)

[2.1 External interfaces 6](#_Toc478326000)

[2.2 Internal structure 6](#_Toc478326001)

[2.3 Roles and responsibilities 7](#_Toc478326002)

[3. Managerial process plans 8](#_Toc478326003)

[3.1 Work plan 8](#_Toc478326004)

[3.2 Work activities 8](#_Toc478326005)

[3.3 Schedule allocation 9](#_Toc478326006)

[4. Technical process plans 10](#_Toc478326007)

[4.1 Process model 10](#_Toc478326008)

[4.2 Methods, tools, and techniques 12](#_Toc478326009)

[5. Risk management plan 13](#_Toc478326010)

[6. References 14](#_Toc478326011)

[Appendix A: Use Case / Scenario Diagrams 15](#_Toc478326012)

[Use Case: Create user 15](#_Toc478326013)

[Use Case: Create Exercise Activity 16](#_Toc478326014)

[Use Case: Add Exercise Record to Exercise Log 17](#_Toc478326015)

[Use Case: Create Exercise Plan 18](#_Toc478326016)

[Use Case: View report 19](#_Toc478326017)

# Preface

The purpose of this document is to define the processes, responsibilities, functions, and metrics for the FitTrac System software. It is intended to be read and understood by all members of the project, and should be referred to when discussing aspects of the project.

# Overview

The FitTrac System will provide user-tailored software for tracking fitness data.

This document is the authoritative document for the FitTrac Software project. It defines the functions, documentation, testing and processes to achieve this. The latest copy of this document will serve as the measure to which these objectives are to be judged as having met product requirements. This document will be maintained by the Project Manager.

# 1. Project Summary

## 1.1 Purpose, scope, and objectives

The purpose of the FitTrac project is to develop a small fitness tracking program (system) that will allow users to tailor the program to track their workout activities, determine the metrics they wish to track, tailor analysis of their workouts, and store workout information. The system enables users to add specified workout descriptions, define data being tracked for those activities, and perform analytical functions of user selected workouts. It will provide the user with a user interface that they may enter User information, create Exercise Activities, and from these be able to apply analysis to user selected data metrics to assess progress.

When successfully completed, the software will enable the user to be able to store user created activities and metrics directly from the user interface. The finished software product will provide a menu and form driven interface for the user to create, enter, view and modify the exercise activity data.

### 1.1.1 Required Features

* Create User Profile
* Create Exercise Activities
* Modify Exercise Activities
* Add Exercise Records to the log based on an Exercise Activity
* Modify Exercise Records
* Delete Exercise Records
* Build an Exercise Plan containing Exercise Activities
* Modify an Exercise Plan
* Delete an Exercise Plan
* Export the Exercise Plan for saving, printing, or importing to a different user profile
* Add all items in an Exercise Plan to the Exercise Log
* Generate charts, graphs, or reports based on the Exercise Log
* Save all Exercise Activities, Logs, and Plans to the user profile
* Load the full user profile

## 1.2 Assumptions and constraints

FitTrac will operate on a desktop computer running Java SE8. The human resources required will be a project manager, a software architect, a design lead programmer, and a software tester. The software is to be written in a java series language with a Swing UI. The database will be in a format compatible with Java, such as MySQL or Apache Derby. The expected length of the project from the release of version 1.0 of this document to completion is approximately 7 weeks.

Hardware constraint:

* Computer running OS X 10.8, Windows 8, Linux version 3.0, or later OS.

Software constraint:

* Java SE8

## 1.3 Project deliverables

The software product delivered will be a series of files containing the software, instructions on its use, and other documentation. The deliverables will include:

* Project Plan document
* Project Design specification
* Test Plan document,
* Beta Phases 1-3
* Final Software Version

During the development phases, input from the customers will be considered for alternative project ideas and additional functionalities.

The software will contain/perform the following:

* User interface for managing fitness scheduling and reports
* Forms for user input
  + Exercise Activity creation
  + Instance creation
  + Report generation
* Database for storing all user input and reports

# 2. Project organization

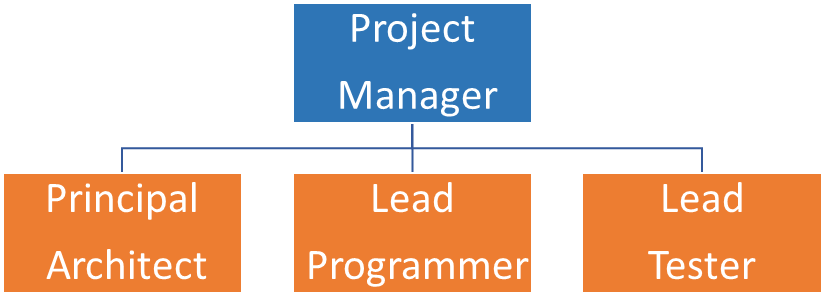


Figure 1 - Team Member Organization

The Project Manager will be the point of coordination and will have final say over project decisions. The Project Manager is responsible for final determination specifications are met. Other team members will be the lead of each of their respective areas with other team members providing support as necessary. The Principal Architect is responsible for ensuring the software plan is reasonable, achievable and useable. The Lead Programmer will ensure project software deliverables that meet the required performance and design criteria are produced. The Lead Tester is responsible for designing and performing software tests are appropriate and timely and results are made available quickly. All team members are responsible for ensuring continued familiarity with this document and for contacting the Project Manager in the event of any conflict that might arise from inconsistency or ambiguity within this document.

## 2.1 External interfaces

The Project Manager shall be a party to all communications with the customer with regards to this project. In addition, the Project Manager will oversee the deliberation on and implementation of quality suggestions from the customer during their reviews.

## 2.2 Internal structure

The internal structure of the project is linear. Due to the small size of the project, there are only four team members. Google Hangouts will be the primary coordination point for project related discussions. In addition, hosting service GitHub will be used to keep track of the project’s progress, as well as storing the system documents in a web based that will be available for all team members. Team members will make themselves available to support other lead responsibilities and will communicate all relevant information to the Project Manager and the lead of the subject area in question. The Project Manager has facilitated an availability schedule to better coordinate collaboration on major work on deliverables. The team will continue to work around busy schedules to stay ahead of schedule and meet all deadlines.

## 2.3 Roles and responsibilities

**Project Manager**

* Manage and document technical and business requirements
* Oversee development of project plan and deliverables
* Define the work packages, and work products for this project.
* Coordinate and facilitate team schedule
* Track current project status and adherence to deadlines
* Verify project deliverables meet requirements

**Principal Architect**

* Develop and document the software architecture
* Determine the appropriate tools, software, and environment
* Ensure deliverables and documentations meet all requirements
* Resolve and communicate technical problems
* Manage and identify risk and mitigation strategies

**Lead Programmer**

* Develops software using Java language
* Implements appropriate database for project input and storage
* Revises code according to team specifications
* Facilitates version control and team access to code repository

**Lead Tester**

* Develop test procedures
* Perform software testing and evaluation
* Document results of testing and analysis
* Provide results to the team members
* Analyze any necessary revisions from testing results

# 3. Managerial process plans

Project management processes for this project are simple due to the nature of the project. The below sections will detail the process this project will follow and the steps to accomplish them.

## 3.1 Work plan

The work plan will follow the schedule and some processes will be concurrent and will feed the other processes. The project will primarily be a waterfall development process.

## 3.2 Work activities

**Preliminary activities:**

The preliminary activities shall consist of the Project Manager and the team members having a formal meeting to determine the technical specifics of the project requirements. These will be documented by the Project Manager and from these a Pseudocode and preliminary software design will be generated. It is expected that the documentation will be generated as this process is occurring.

Once the Pseudocode and software design have been reviewed to ensure they are meeting the customer’s specification and intent, the design phase will begin.

**Design phase:**

The Design phase shall consist of turning the Pseudocode into software modules. The programmer will make the User Interface design. It will be refined and set to ensure that it is adequate to both the requirements of the customer as well as the database. The front-end software will take into account customer business requirements for reports, views, and changes as well as the completion of the database. Once these three elements (UI, Database, and front end software) are determined to satisfy the customer requirements, technical constraints, as well as practical considerations, a review will be conducted with the Project Manager to ensure all considerations have been made.

**Software development:**

The Software Development shall focus on the development of the program module code and user forms for use in the final project. On completion of all the modules, the programmer will perform initial testing prior to the review.

**Testing phase:**

Testing shall be both black box and white box testing with special care taken to ensure most edge cases are handled. Edge cases that cannot be handled by the system shall be documented for the customer as well as suggestions for process adaptation.

## C:\Users\RECar\AppData\Local\Microsoft\Windows\INetCache\Content.Word\schedule 2.0.png3.3 Schedule allocation

Table 1 - Work Activity Schedule

# 4. Technical process plans

This project utilizes the waterfall development process model, this section describes the technical methods, tools, and techniques to be used to develop the various work products; plans for establishing and maintaining theproject infrastructure; and the product acceptance plan.

## 4.1 Process model

The waterfall development model will be used for the management of this project. The Project Manager may waive portions of this model where the rigidity of adherence would produce adverse outcome. Due to the small staff size of this project, processes may overlap and enter periods of simultaneous development. This flexibility will enable the finished product to be produced at a faster pace with deliverables more specific to the customer’s desires than would be possible with a large project. The process starts with the requirements analysis. This analysis will be used to develop sub-processes and system requirements that will be used to determine the achievement of project milestones. The final milestone, project completion serves as the point at which delivery of a finished, tested product will be delivered to the customer for acceptance testing.

Example data flow models follow:

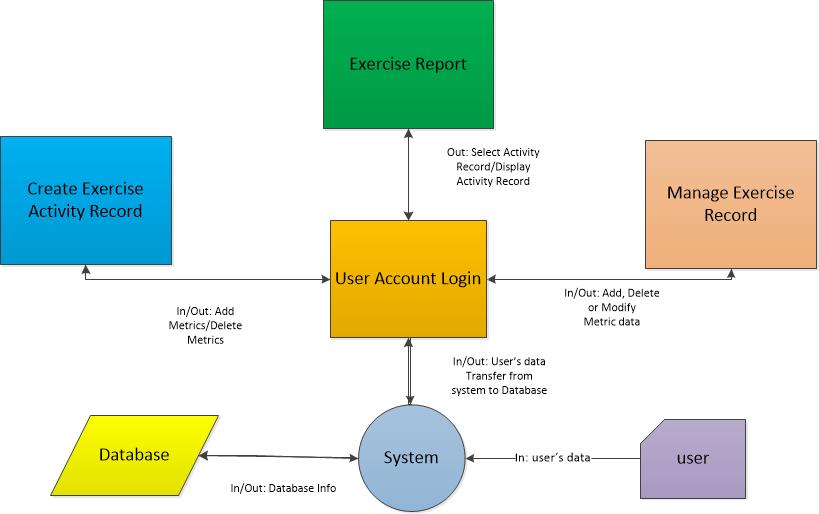


Figure 2 - Level 0 Data Flow Diagram

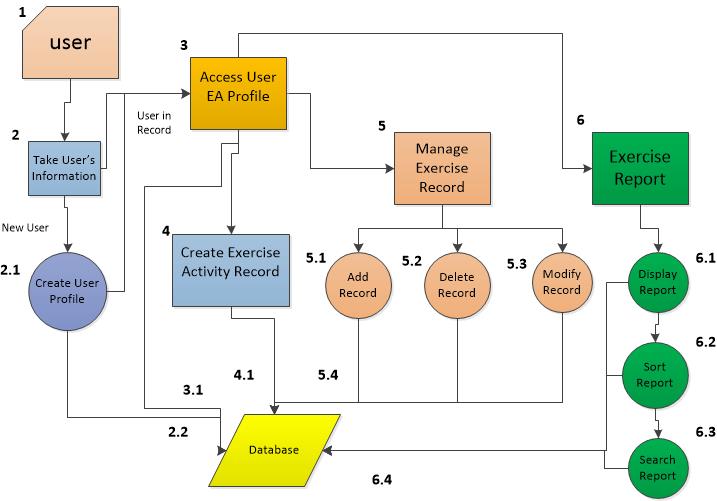


Figure 3 - Level 1 Data Flow Diagram

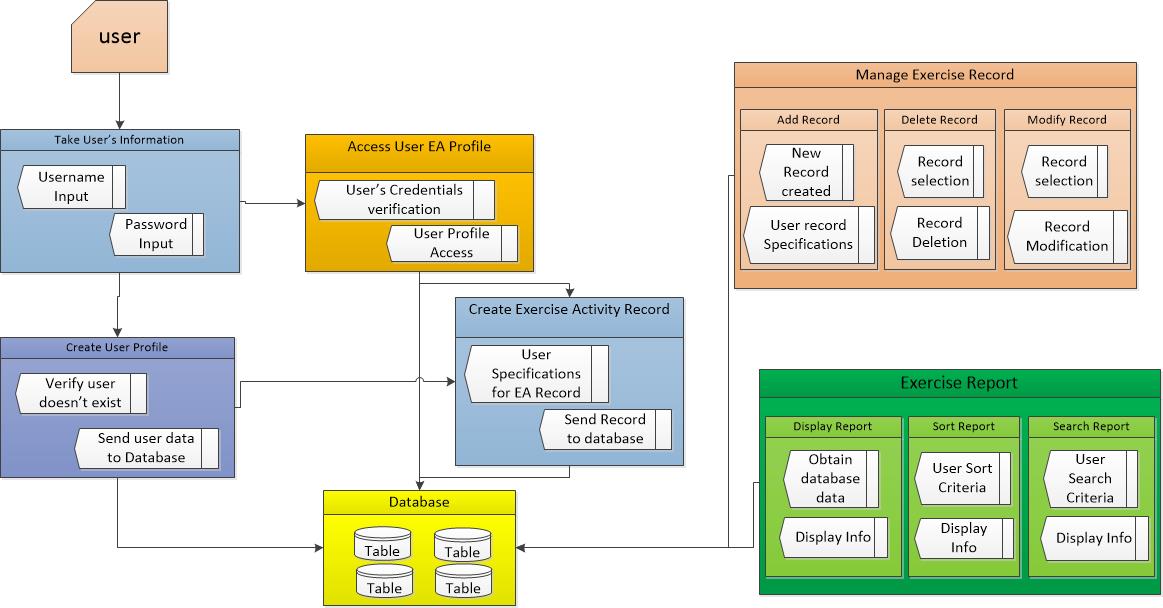


Figure 4 - Level 2 Data Flow Diagram

## 4.2 Methods, tools, and techniques

The project will be following the waterfall methodology for program management. Additional documentation, testing integration and development will utilize IEEE standards for software development. The following tools will be used in the production of this product.

**Languages**

- Java

**IDE and Database software**

- Eclipse/NetBeans

**Documentation**

- Microsoft Office

**Software Repository**

- GitHub: <https://github.com/Xavyor/FitTrac>

# 5. Risk management plan

Risk management for FitTrac is primarily on the correct configuration and performance of the modules to deliver the desired functionality to the customer. The primary risk of the project is the adding additional requirements prior to project conclusion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Priority | Description | Probability | Impact | Mitigation |
| 1 | Customer requires additional data capture/functionality stand alone. | High | low | Design consideration for extensibility |
| 1 | Customer wants a web enabled interface to connect to a website, not in requirements | Medium | High | Design interface for this capability |
| 2 | Customer wants additional features not accounted for with initial SRS | Medium | High | Reevaluate with regards to schedule/customer input |
| 3 | Customer hardware incompatible with software | Low | High | Include hardware requirements as part of documentation. |

Table 2 - Risk Management Table

To account for reasonably anticipated risk, sufficient communication with the customer of the capabilities and limitations of the software proposed for must be made. Specific actions and costs must be associated with any substantive changes to the agreed capabilities of the project being developed, to include cut off dates for acceptable changes.

To this end, the Project Manager will provide the customer with a detailed description of the software in development as well as design constraints based on the software requirements.

# 6. References

*IEEE Standard for Software Project Management Plans*. (1998) (1st ed.). New York.

Retrieved from <https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2678>**.**

# Appendix A: Use Case / Scenario Diagrams

## Use Case: Create user

**Pre-condition:** The user profile does not exist

**Post-condition:** The user profile exists and is loaded

**Actor Profile:** User is opening FitTrac for the first time and inputting basic information

**Sequence of Events:**

1. User opens the program
2. System prompts user to create a new account
3. User fills in basic information, including name, height, and weight
4. System generates a user profile based on that information

**Scenarios**

**User Scenario 1:** User opens program

1. User opens the program
2. System determines which user profiles are available
3. System prompts the user to create a new profile since no profiles are available

**User Scenario 2:** User information is entered

1. System presents the user creation form to the user
2. User enters name
3. User enters height
4. User enters weight
5. User confirms by selecting the Create button
6. System creates the user profile
7. System creates an Exercise Activity table for the user and fills it with default activities
8. System creates an empty Exercise Log

## Use Case: Create Exercise Activity

**Pre-condition:** The user has a profile and is in the main menu for it

**Post-condition:** The Exercise Activity has been created

**Actor Profile:** User adds a new Exercise Activity to the profile

**Sequence of Events:**

1. User decides to add a new Exercise Activity
2. User provides required information
3. System verifies the information and adds it to the table

**Scenarios**

**User Scenario 1:** User selects Add Exercise Activity menu option

1. System displays main menu
2. User selects Add Exercise Activity from the menu
3. System prompts the user for a type of exercise (Cardio, Strength, etc.)
4. User selects exercise type

**User Scenario 2:** User selects the Cardio exercise type

1. System prompts the user for Exercise Activity generic and Cardio specific information
2. User enters exercise name
3. User enters other information
4. User selects Create
5. System verifies the information and any key conflicts
6. System creates the Exercise Activity

## Use Case: Add Exercise Record to Exercise Log

**Pre-condition:** The user has already entered the Exercise Activity

**Post-condition:** The user has logged an instance of that Exercise Activity

**Actor Profile:** The user has completed an exercise and wants to log it.

**Sequence of Events:**

1. User completes an exercise
2. User logs the exercise

**Scenarios**

**User Scenario 1:** User selects an Exercise Activity

1. The user selects the Log option from the main menu
2. The system displays a list of Exercise Activities
3. The user selects from the list and confirms by pressing a button

**User Scenario 2:** User updates the log

1. The system prompts the user for information specific to the exercise type
2. The user fills in information about the activity, including quantity (time, distance, repetitions, etc), and comments.
3. The user confirms the log entry by pressing a button
4. The system logs the instance of the Exercise Activity

## Use Case: Create Exercise Plan

**Pre-condition:** The user has already created multiple Exercise Activities

**Post-condition:** The user has created an Exercise plan

**Actor Profile:** The user wants to plan an exercise session

**Sequence of Events:**

1. The user has an idea for which exercises should be part of a session
2. The user selects those Exercise Activities and adds a projected quantity (time, distance, repetitions, etc.) for the activity

**Scenarios**

**User Scenario 1:** User adds an Exercise Activity to the Plan

1. The system displays all the Exercise Activities
2. The user selects an Exercise Activity
3. The system prompts the user for a quantity
4. The user enters a quantity
5. The system adds the Exercise Activity to the plan

**User Scenario 2:** User completes the Exercise Plan

1. The user reviews the list of exercises and quantities
2. The user clicks the save button
3. The system stores the Exercise Plan
4. The system prompts the user with an option to export the Exercise Plan

## Use Case: View report

**Pre-condition:** The user has multiple exercises logged that are similar enough to be compared

**Post-condition:** The system displays a report

**Actor Profile:** User wants to compare past performance of certain exercises

**Sequence of Events:**

1. User selects the reports
2. User selects which report to display
3. System displays that report

**Scenarios**

**User Scenario 1:** User selects Running report

1. User selects Reports from the main menu
2. System displays the list of possible reports
3. User selects Running from the list
4. System displays a report of previous runs with options to compare by distance or time

**User Scenario 2:** User selects Weight Lifting report

1. User selects Reports from the main menu
2. System displays the list of possible reports
3. User selects Weight Lifting from the list
4. System displays a report of previous lifts with options to compare by weight and repetitions

Current Version Approval

|  |  |  |
| --- | --- | --- |
| **Approved by** | **Date** | **Signature** |
| PM, Ryan Carnes | 3/25/17 | REC |
| PM, Ryan Carnes | 3/26/17 | REC |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Change History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No.** | **Pages changed** | **Change Description** | **Approval** | **Release Date** |
| 1.0 | All | Initial Release, AS, ABR | REC | 3/25/17 |
| 1.1 | ii, 9, 15-19 | Update to Initial Release, GF | REC | 3/26/17 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |