IT425-1701B-01: Systems Analysis, Design, and Integration

Project Name: FitTrac©

Michael J Davis

March 22, 2017



Table of Contents

[Week 1: System or Application Overview 5](#_Toc477854947)

[Capabilities 5](#_Toc477854948)

[Stakeholders 5](#_Toc477854949)

[Sponsor 5](#_Toc477854950)

[Week 1: Requirements Specification 6](#_Toc477854951)

[FitTrac goals and objectives 6](#_Toc477854952)

[Requirements elicitation methods 6](#_Toc477854953)

[Capturing Requirements 6](#_Toc477854954)

[Functional Requirements (RTM) 7](#_Toc477854955)

[Non-Functional Requirements (RTM) 11](#_Toc477854956)

[Week 2: System or Application Design 14](#_Toc477854957)

[Use Case Diagram 14](#_Toc477854958)

[Functional Decomposition Diagram 15](#_Toc477854960)

[Log-In Data Flow Diagram 16](#_Toc477854961)

[Goal Manager Data Flow Diagram 17](#_Toc477854962)

[Activity Manager Data Flow Diagram 18](#_Toc477854963)

[Nutrition Manager Data Flow Diagram 19](#_Toc477854964)

[Fitness Summary Manager Part I – Vital Stats Data Flow Diagram 20](#_Toc477854965)

[Fitness Summary Manager Part II – Goals & Summaries Data Flow Diagram 21](#_Toc477854966)

[Profile Picture Manager Data Flow Diagram 22](#_Toc477854967)

[Random Quote & Picture Manager Data Flow Diagram 23](#_Toc477854968)

[Sequence Diagram 24](#_Toc477854969)

[High-Level UI (Log-in) 25](#_Toc477854970)

[High-Level UI (Dashboard) 26](#_Toc477854971)

[Week 3: Test and Quality Assurance Plan 27](#_Toc477854972)

[Quality Assurance Plan 27](#_Toc477854973)

[Introduction 27](#_Toc477854974)

[QA Team Selection 28](#_Toc477854975)

[Scope 28](#_Toc477854976)

[Unit Test Plan 42](#_Toc477854977)

[System Testing Plan 46](#_Toc477854978)

[User Acceptance Test Plan 59](#_Toc477854979)

[Week 4: Development Strategy 64](#_Toc477854980)

[Development Plan Outline 64](#_Toc477854981)

[Outsourcing Proposal 64](#_Toc477854982)

[Insourcing Proposal 73](#_Toc477854983)

[Combination Proposal 78](#_Toc477854984)

[Development Strategy Recommendation 86](#_Toc477854985)

[Week 5: Integration and Development Plan 87](#_Toc477854986)

[Work Breakdown Structure – Preliminary Analysis 87](#_Toc477854987)

[Work Breakdown Structure – Requirements Definition 88](#_Toc477854988)

[Work Breakdown Structure – System Design 89](#_Toc477854989)

[Work Breakdown Structure – Development 90](#_Toc477854990)

[Work Breakdown Structure – Integration and Testing 91](#_Toc477854991)

[Work Breakdown Structure – Acceptance, Installation, and Deployment 92](#_Toc477854992)

[Work Breakdown Structure – Operation Support and Maintenance 93](#_Toc477854993)

[References 94](#_Toc477854994)

# Week 1: System or Application Overview

## Capabilities

The application slated for creation will be a personal fitness management system called ***FitTrac***. FitTrac is a web application that NextGen Fitness will provide as a service to their clientele as a goal setting and tracking tool. FitTrac will allow users to input their workout data and compare their current progress to their goal and review summaries on how well they are performing. The application will host a feature for client’s to view custom meal plans that have been created and uploaded by their fitness trainer. Personal trainers and other managers will be able to log into the system and assist their clients while they are away from the gym with all aspects of their fitness goals. Gym members who do not utilize personal training services will still be able to track their own progress, but will not have the added feature of online fitness coaching.

## Stakeholders

Primary stakeholders: (Editorial Board, 2016)

* End-users (Franchise clientele)
* Sponsor (NextGen Fitness)
* Project Manager (Crispin Jose)
* Developers
* Technical Writers
* Testers

Secondary stakeholders: (Editorial Board, 2016)

* Sales and Marketing team
* Accounting team

## Sponsor

The project will be sponsored by ***NextGen Fitness****,* a fitness franchise that is seeking to create a fitness management system for their clientele and expand their market share through the use of proven fitness techniques and current technology.

# Week 1: Requirements Specification

## FitTrac goals and objectives

* Provide an easy to use system that augments NextGen Fitness clientele training sessions by providing support during their times apart from their trainer
* Create a productivity application that becomes synonymous with the NextGen Fitness mission of “*Fit for the Future*”
* Increase clientele retention rates through tracking activities that clients as well as trainers can view and work together toward achieving the client’s fitness goals

## Requirements elicitation methods

1. Workshop held with the project manager, developers, secondary stakeholders, and NextGen Fitness project POCs such as directors, managers, and master trainers
2. Questionnaire for members and clients
3. Brainstorming session
4. Use Case modeling
5. Requirements Traceability Matrix (RTM) documentation

## Capturing Requirements

Both functional and non-functional requirements will be entered within a *requirements traceability matrix* (*RTM*) (Editorial Board, 2016). The RTM will not only capture requirements, but also be used as a reference during later phases of the SDLC (Editorial Board, 2016).

## Functional Requirements (RTM)

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT001** |
| Requirement Statement | ***The system shall support up to 500 concurrent users per server.*** |
| Priority | 3 |
| Source | NextGen Fitness |
| Risk | Application downtime due to server overload |
| Opportunity | Support for high concurrent server traffic |
| Category | Functional - Engineering |
| Test Acceptance Criteria | Successful simulation of 500 concurrent users per server |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT002** |
| Requirement Statement | ***The system shall support the creation, editing, and deletion of up to five fitness goals and their corresponding milestone dates as outlined during trainer coaching sessions.*** |
| Priority | 8 |
| Source | NextGen Fitness |
| Risk | Limited functionality and value for applications users |
| Opportunity | Increased application value for users |
| Category | Functional - Engineering |
| Test Acceptance Criteria | Successful creation, editing, and deletion of five sample fitness goals and their corresponding milestone dates |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT003** |
| Requirement Statement | ***The system shall support the addition and view of fitness activities and their corresponding information by both clients and trainers.*** |
| Priority | 9 |
| Source | NextGen Fitness |
| Risk | Limited functionality and degraded system ability to track activity versus progress toward goal(s) attainment. |
| Opportunity | Enhanced tracking of fitness activities and their impact on overall goal attainment |
| Category | Functional - Engineering |
| Test Acceptance Criteria | Successful addition and view of sample fitness activities and their corresponding information through client and trainer log-in |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT004** |
| Requirement Statement | ***The system shall support user view, export, and print of up to three custom menu plans that are uploaded by their fitness trainer as outlined during coaching sessions.*** |
| Priority | 10 |
| Source | NextGen Fitness |
| Risk | Limited functionality and Major loss of application value due to the exclusion of a vital pillar of fitness goal attainment |
| Opportunity | Enhanced tracking of client food intake and supplementation in support of fitness goal attainment |
| Category | Functional - Engineering |
| Test Acceptance Criteria | Successful upload of up to three custom menu plans through trainer log-in and successful view, export, and print of uploaded plans through client and trainer log-in |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT005** |
| Requirement Statement | ***The system shall provide summary views of fitness goal attainment progress and client vital stats. The system shall allow the view and editing of vital stats and the view and export of fitness goals summaries.*** |
| Priority | 11 |
| Source | NextGen Fitness |
| Risk | Limited functionality and major loss of application value for clients that want to view and share their progress with others |
| Opportunity | Increased application value for users; potential marketing tool through client affiliate interest in program effectiveness |
| Category | Functional - Engineering, Non-Functional - Sales and Marketing |
| Test Acceptance Criteria | Successful view and export of sample client progress summaries and successful view and edit of vital stat history. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT006** |
| Requirement Statement | ***The system shall provide a feature for clients to upload, change, and delete an image of their choosing to serve as a profile picture.*** |
| Priority | 12 |
| Source | NextGen Fitness |
| Risk | Minimal loss of application value |
| Opportunity | Minor increased value through the addition of a user customization option |
| Category | Functional - Minor extra features |
| Test Acceptance Criteria | Successful upload, change, and deletion of a sample image through the client log-in |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT007** |
| Requirement Statement | ***The system shall provide a feature that will cycle through a list of up to 50 random inspirational quotes to provide client motivation.*** |
| Priority | 14 |
| Source | Development team |
| Risk | Minimal to no loss of application value |
| Opportunity | Minor increased value through the addition of a motivational tools |
| Category | Functional - Minor extra features |
| Test Acceptance Criteria | Successful upload and cycle through a list of 50 inspirational quotes. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT008** |
| Requirement Statement | ***The system shall provide a feature that will cycle through a list of up to 10 random NextGen Fitness related images to provide client motivation and reinforce branding.*** |
| Priority | 13 |
| Source | Sales and Marketing team |
| Risk | Minor loss of application value; Major loss of NextGen Fitness brand reinforcement |
| Opportunity | Minor increased value through the addition of a motivational tool; Major increased value through brand reinforcement |
| Category | Functional - Major extra features |
| Test Acceptance Criteria | Successful upload and cycle through a list of 10 NextGen Fitness related images. |

## Non-Functional Requirements (RTM)

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT009** |
| Requirement Statement | ***The system shall be accessible only via a registered and properly authenticated client, trainer, or manager account log-in*** |
| Priority | 4 |
| Source | Development team |
| Risk | Major security vulnerability of personal client information, NextGen Fitness business rules, and development company code |
| Opportunity | Increased application security measures for stakeholders |
| Category | Non-Functional - Security |
| Test Acceptance Criteria | Successful registration and log-in of test client, trainer, and manager accounts. Unsuccessful log-in and temporary lock-out after four failed attempts at account authentication. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT010** |
| Requirement Statement | ***The system shall support PC and Mac OS users.*** |
| Priority | 6 |
| Source | NextGen Fitness |
| Risk | Compatibility issues for users with unsupported OS |
| Opportunity | Access to larger user base due to multiple OS compatibility |
| Category | Non-Functional - Usability |
| Test Acceptance Criteria | Successful application usage with PC and Mac OS supported systems. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT011** |
| Requirement Statement | ***The system shall be a web application designed with 3-tier architecture - utilizing technologies such as HTML, PHP, JavaScript, MySQL, and Java.*** |
| Priority | 5 |
| Source | Source: Development team |
| Risk | Lack of scalability will create a shortened overall product life cycle |
| Opportunity | Scalability for future versions and cost decrease through use of open source technologies |
| Category | Technology (assumptions) |
| Test Acceptance Criteria | Successful system deployment and use in a 3-tiered architectural environment. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT012** |
| Requirement Statement | ***The system shall not provide menu planning and trainer support for users who are not actively utilizing personal training services.*** |
| Priority | 7 |
| Source | NextGen Fitness |
| Risk | Loss of sponsor revenue due to members having access to features that will decrease the need for personal training services. |
| Opportunity | Promotion of the menu planning and trainer support features as part of the purchase of personal training services. |
| Category | Non-Functional - Major extra feature (assumptions) |
| Test Acceptance Criteria | Client access to menu and trainer support features. Lack of access to menu and trainer support features for members not utilizing personal training services. |

|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT013** |
| Requirement Statement | ***The system shall be fully-functional and deployed within an 18 month time frame.*** |
| Priority | 2 |
| Source | NextGen Fitness, Project Manager, Accounting team |
| Risk | Loss of sponsor revenue and clientele increase, Negative impact to development company reputation. |
| Opportunity | To become the preferred development company for sponsor solution needs and positive impact to development company reputation. |
| Category | Non-Functional - Timetables (constraints) |
| Test Acceptance Criteria | Fully-functional system delivered on or before the 18 month deadline. |

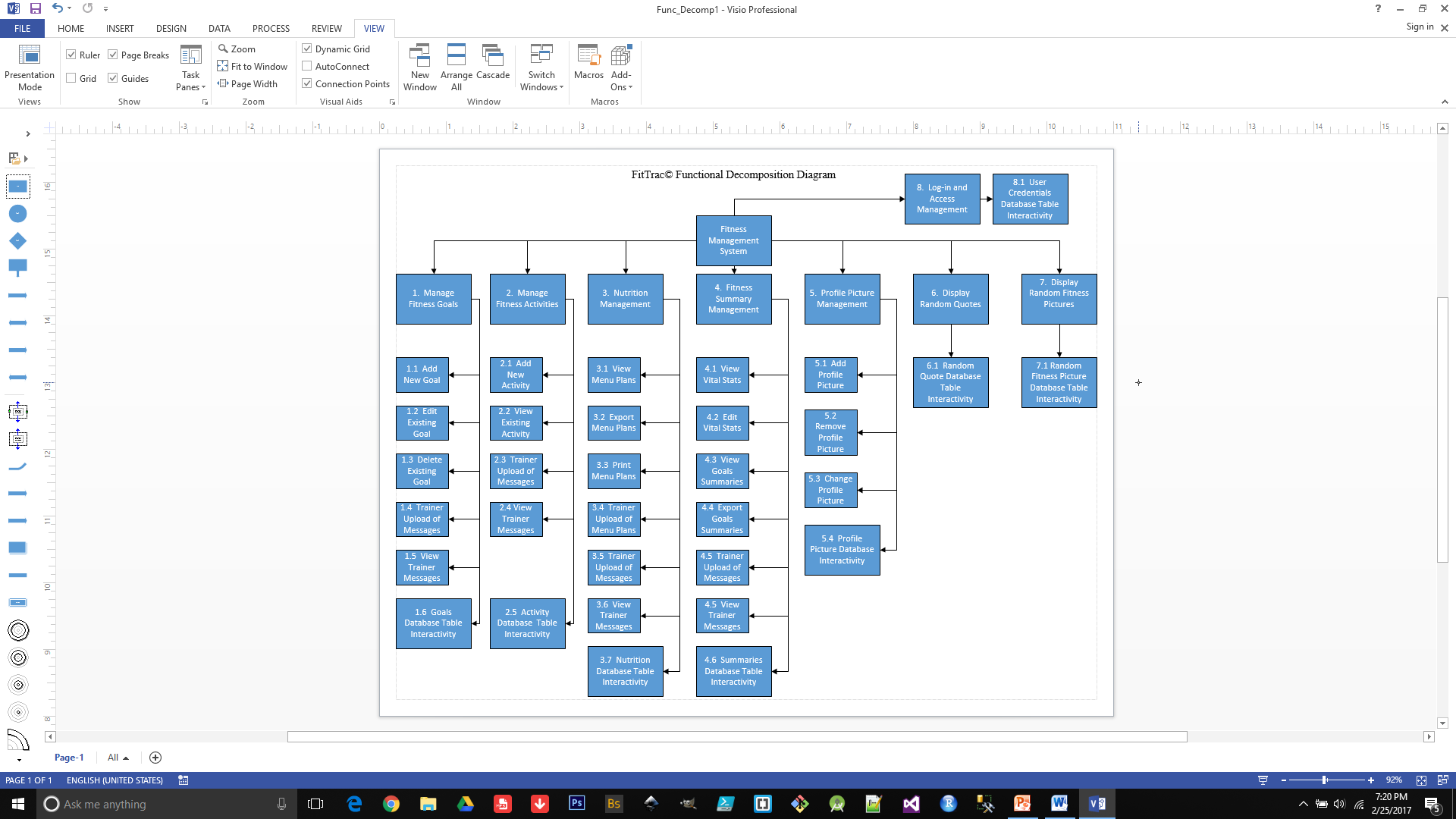
|  |  |
| --- | --- |
| RTM FIELD | DESCRIPTION |
| ID | **FIT014** |
| Requirement Statement | ***The system shall be completed within an allotted project budget of $800,000.*** |
| Priority | 1 |
| Source | NextGen Fitness, Project Manager, Accounting team |
| Risk | Increased costs for sponsor, negative impact to development company reputation |
| Opportunity | To become the preferred development company for sponsor solution needs and positive impact to development company reputation. |
| Category | Non-Functional - Financials (constraints) |
| Test Acceptance Criteria | Fully-functional system delivered under the $800,000 budget |

# Week 2: System or Application Design

## Use Case Diagram

### 

## Functional Decomposition Diagram



Return Comparison Results

Check Credentials against Database

Log-In Message. On Success, Manager Access

Send Log-In Credentials

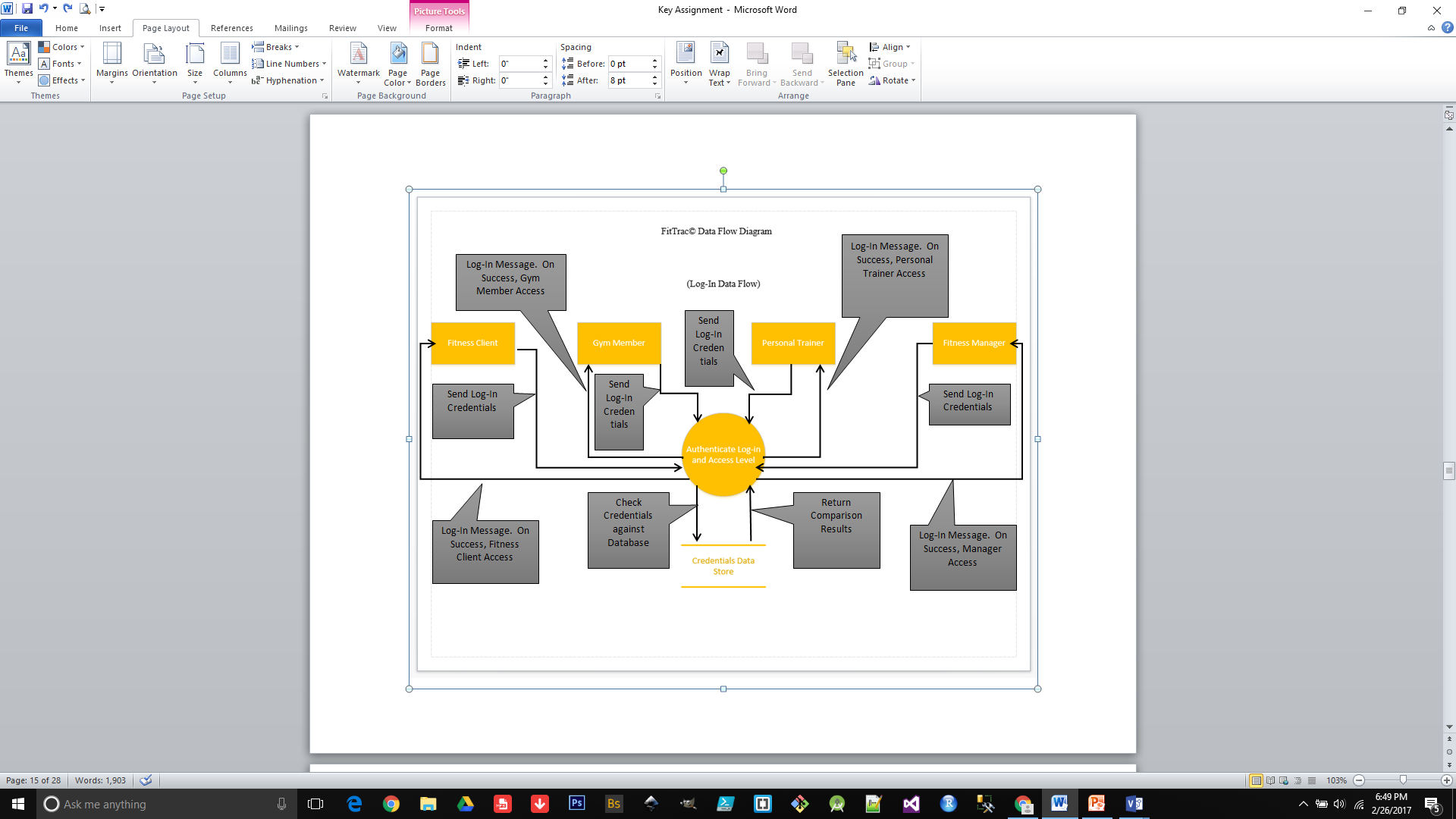
Send Log-In Credentials

Log-In Message. On Success, Fitness Client Access

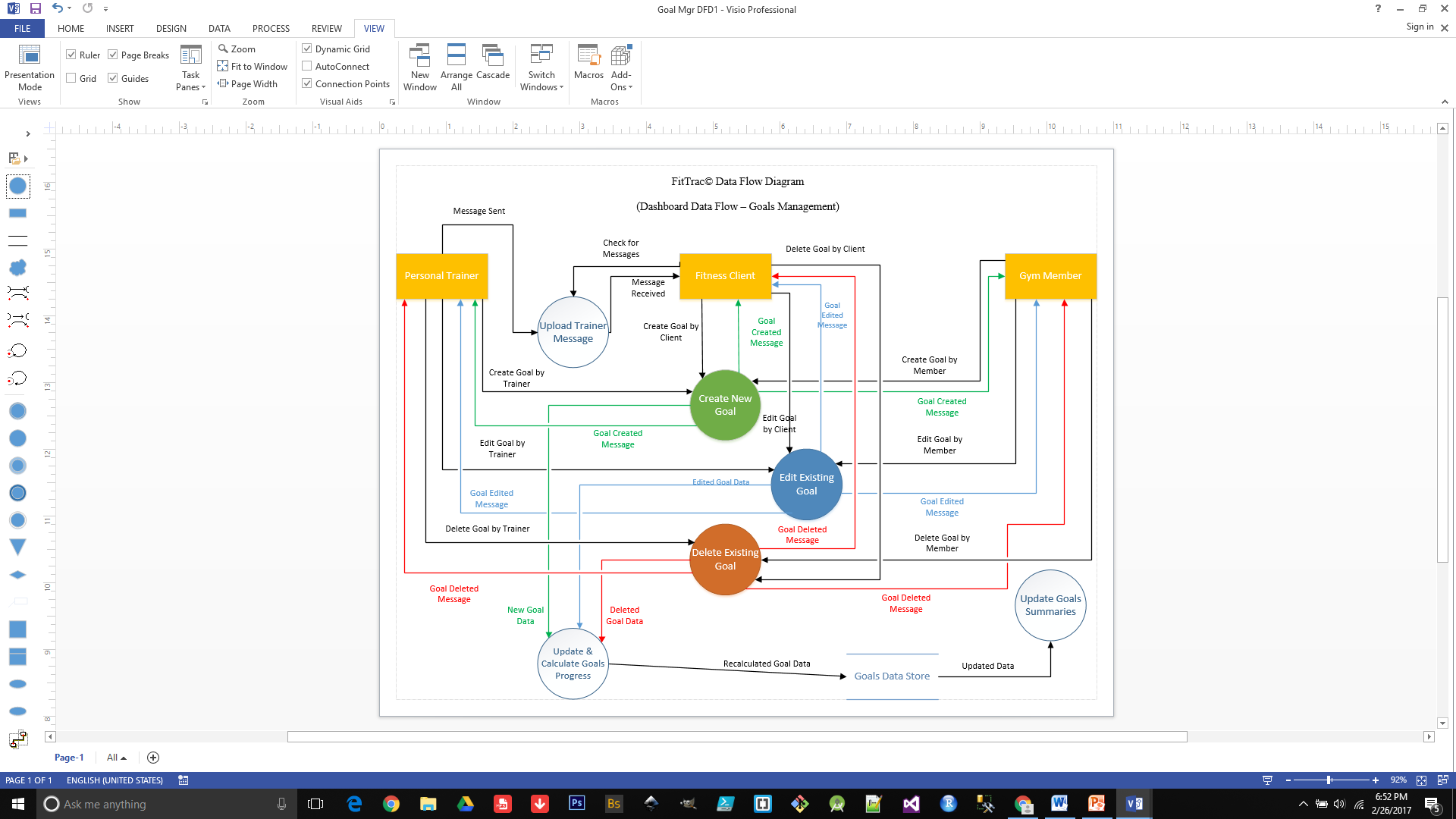
Send Log-In Credentials

Send Log-In Credentials

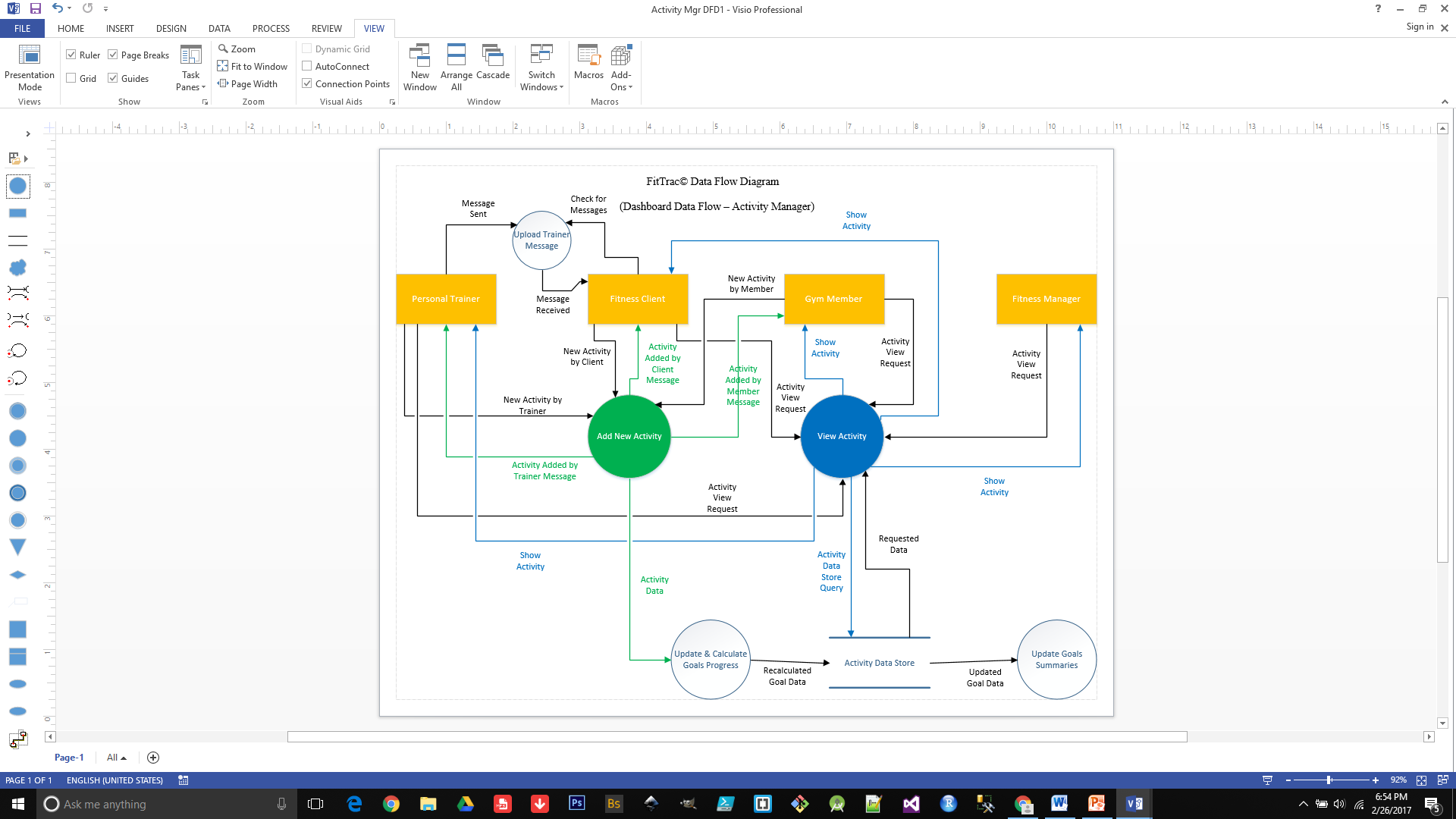
## Log-In Data Flow Diagram



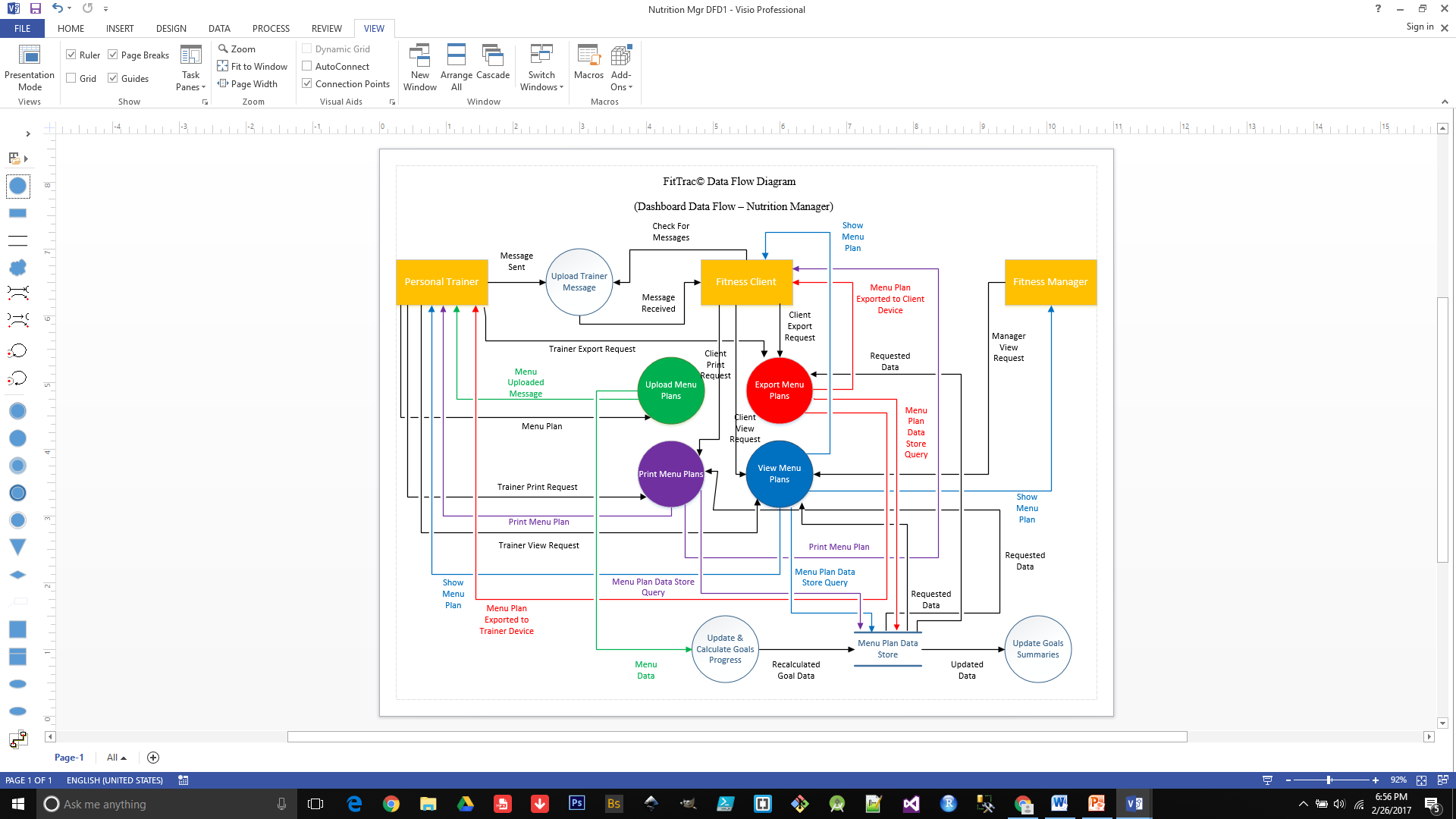
## Goal Manager Data Flow Diagram



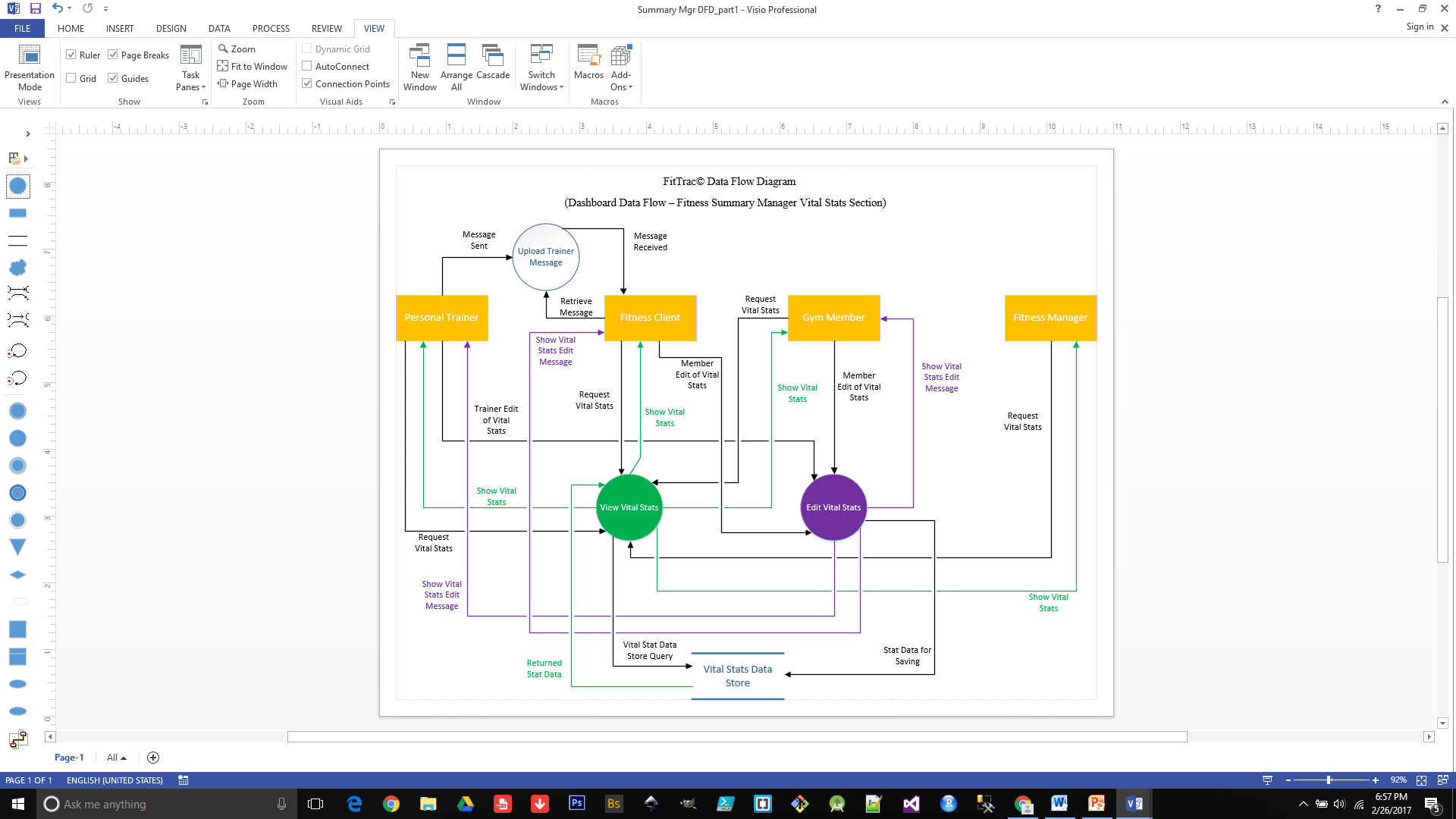
## Activity Manager Data Flow Diagram



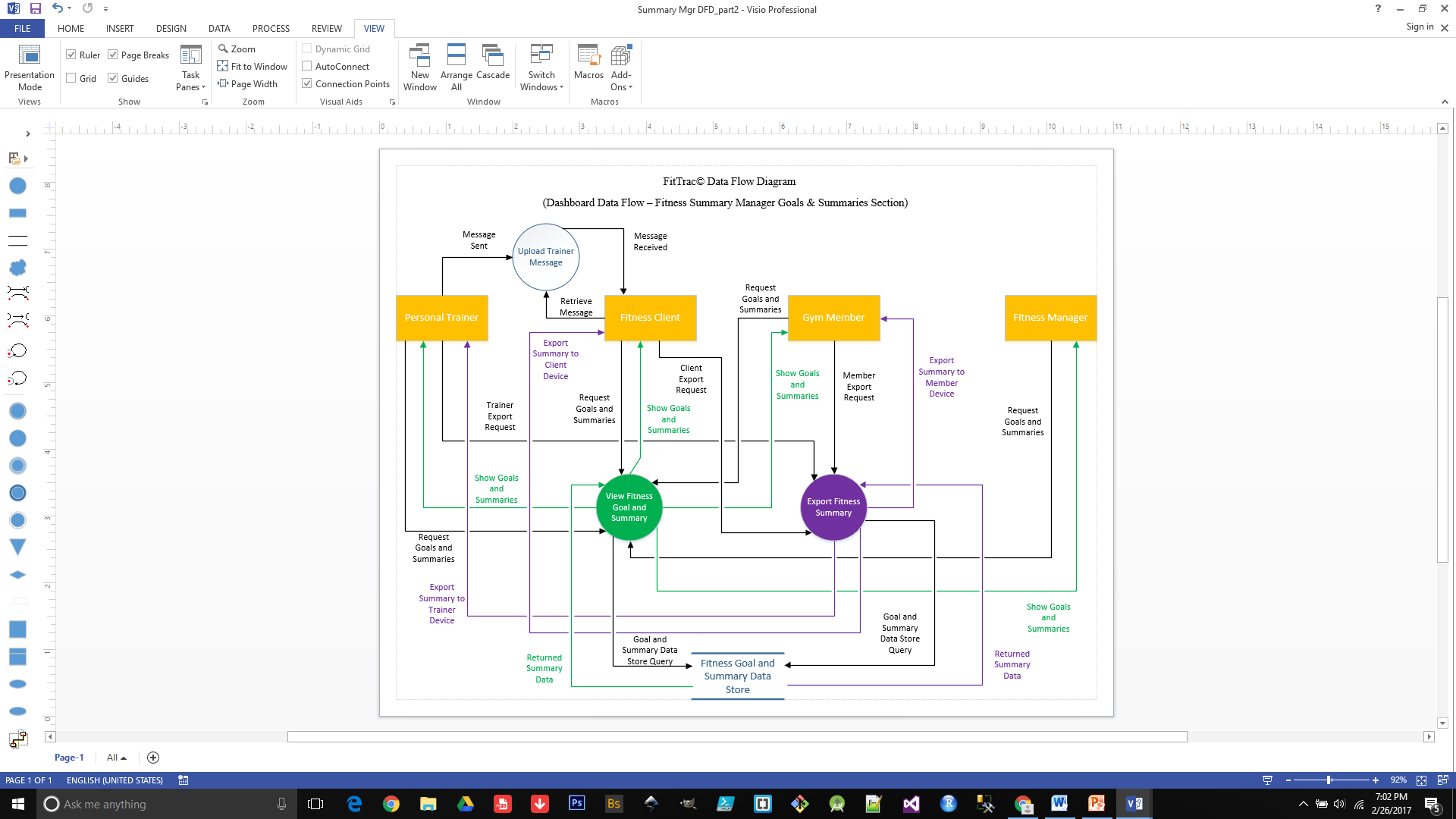
## Nutrition Manager Data Flow Diagram



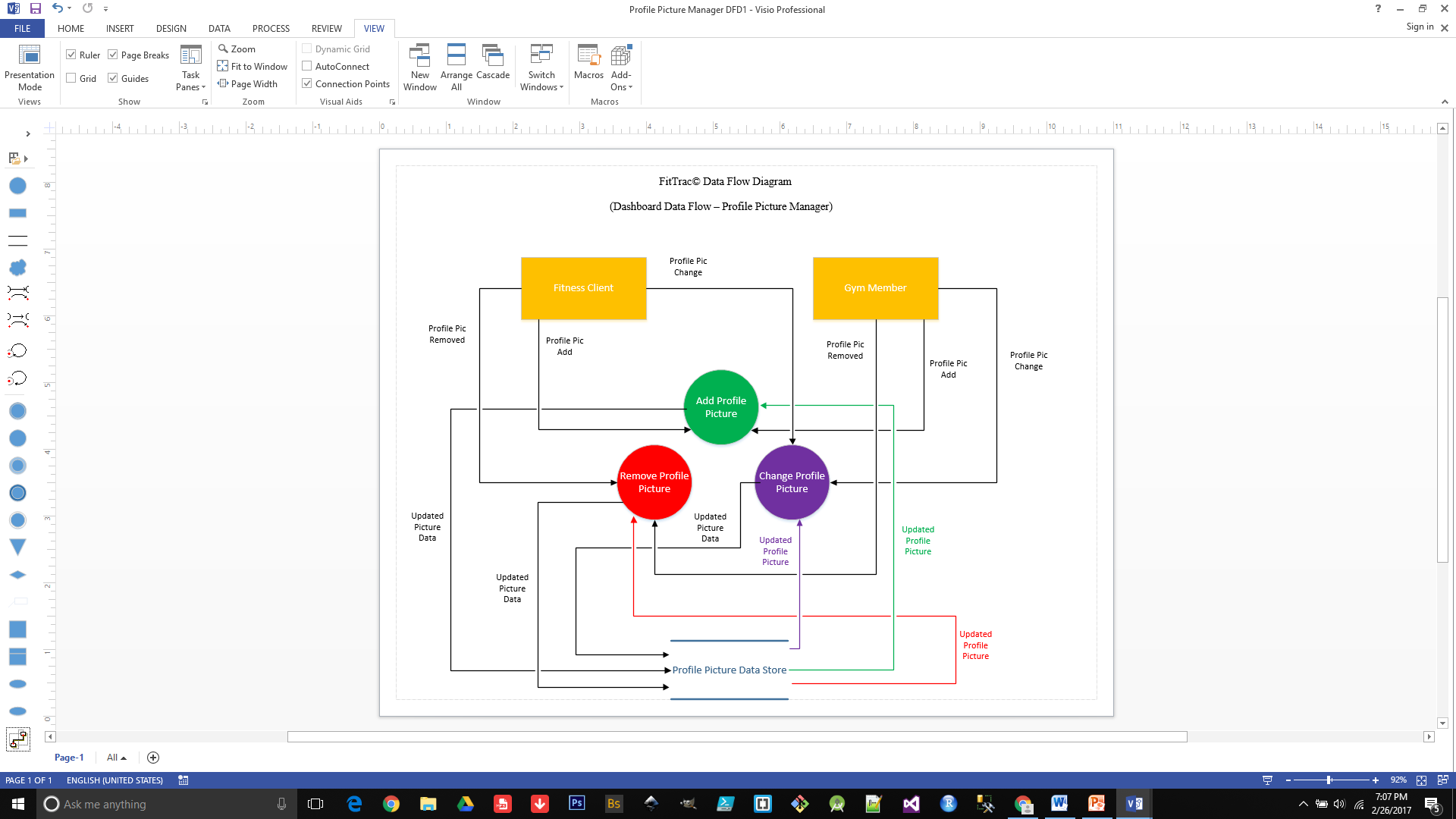
## Fitness Summary Manager Part I – Vital Stats Data Flow Diagram



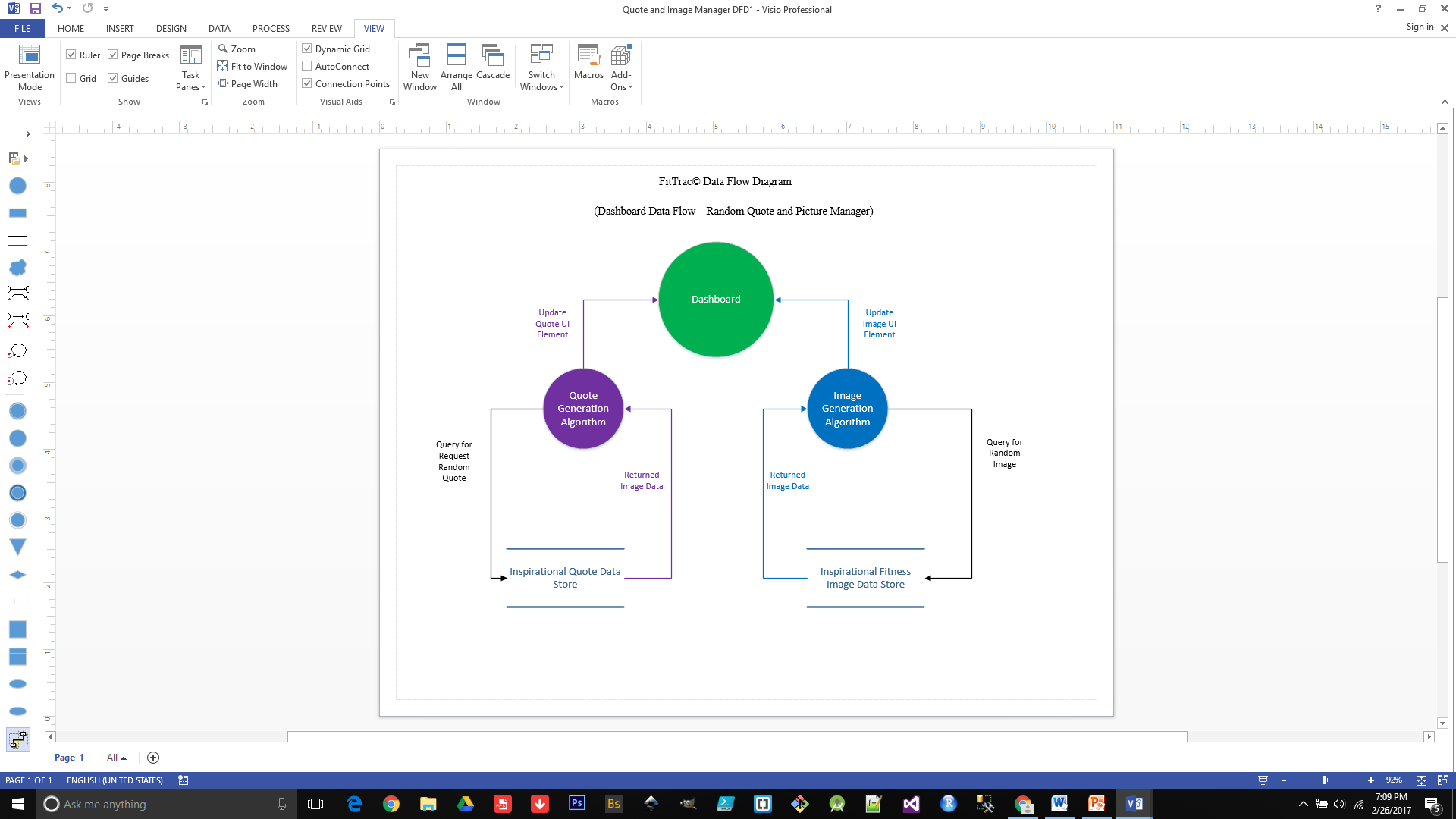
## Fitness Summary Manager Part II – Goals & Summaries Data Flow Diagram



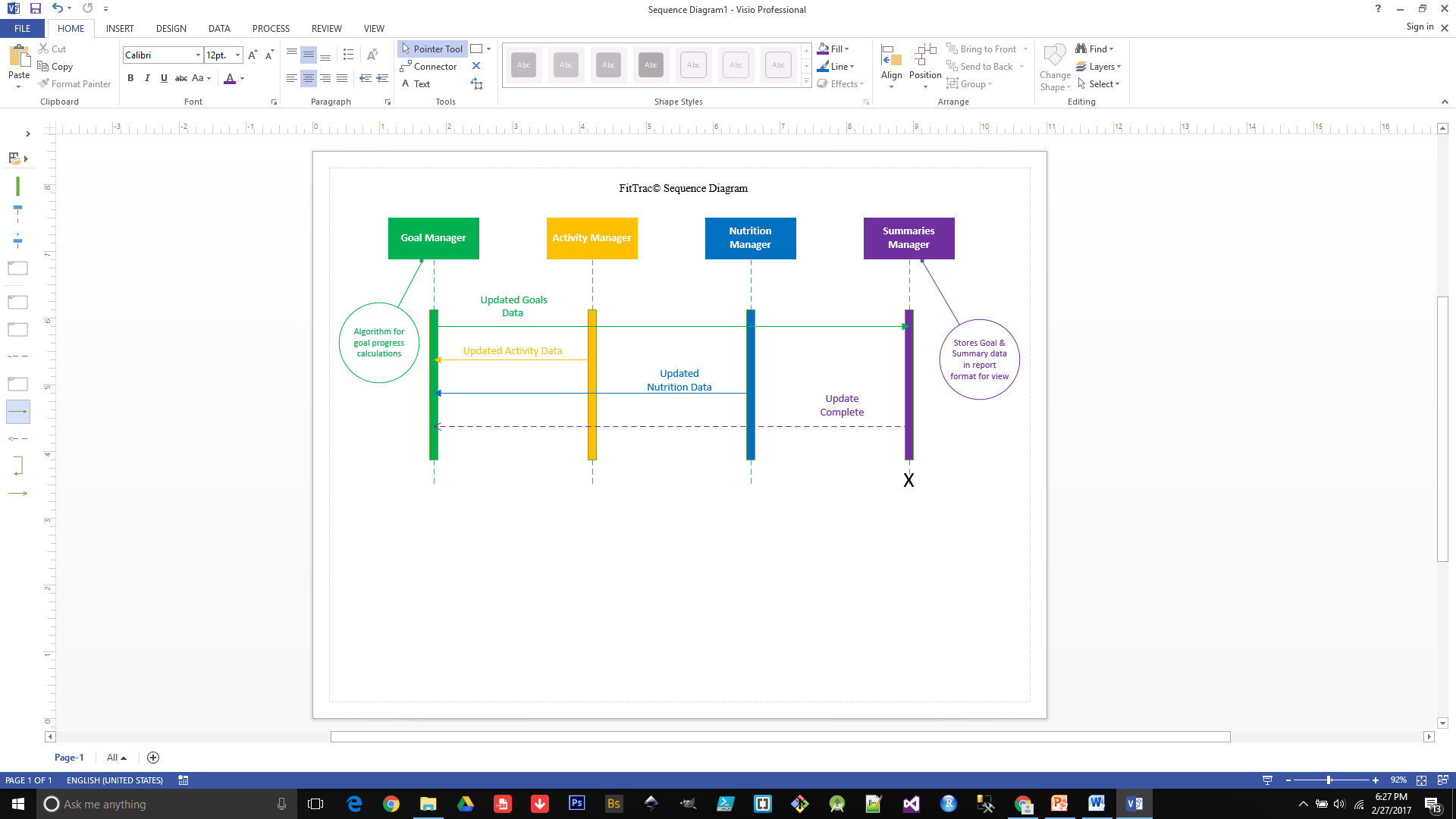
## Profile Picture Manager Data Flow Diagram



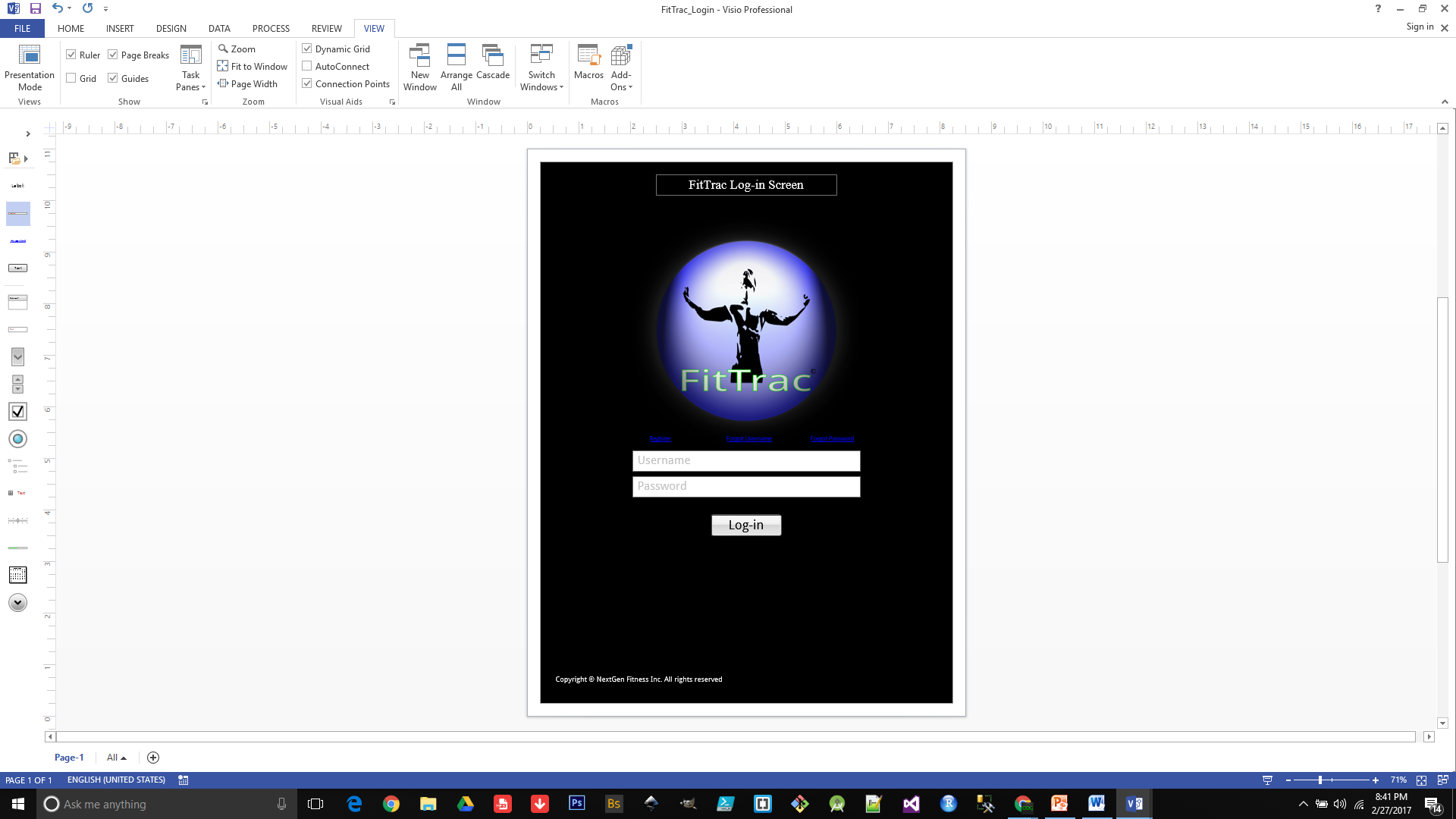
## Random Quote & Picture Manager Data Flow Diagram



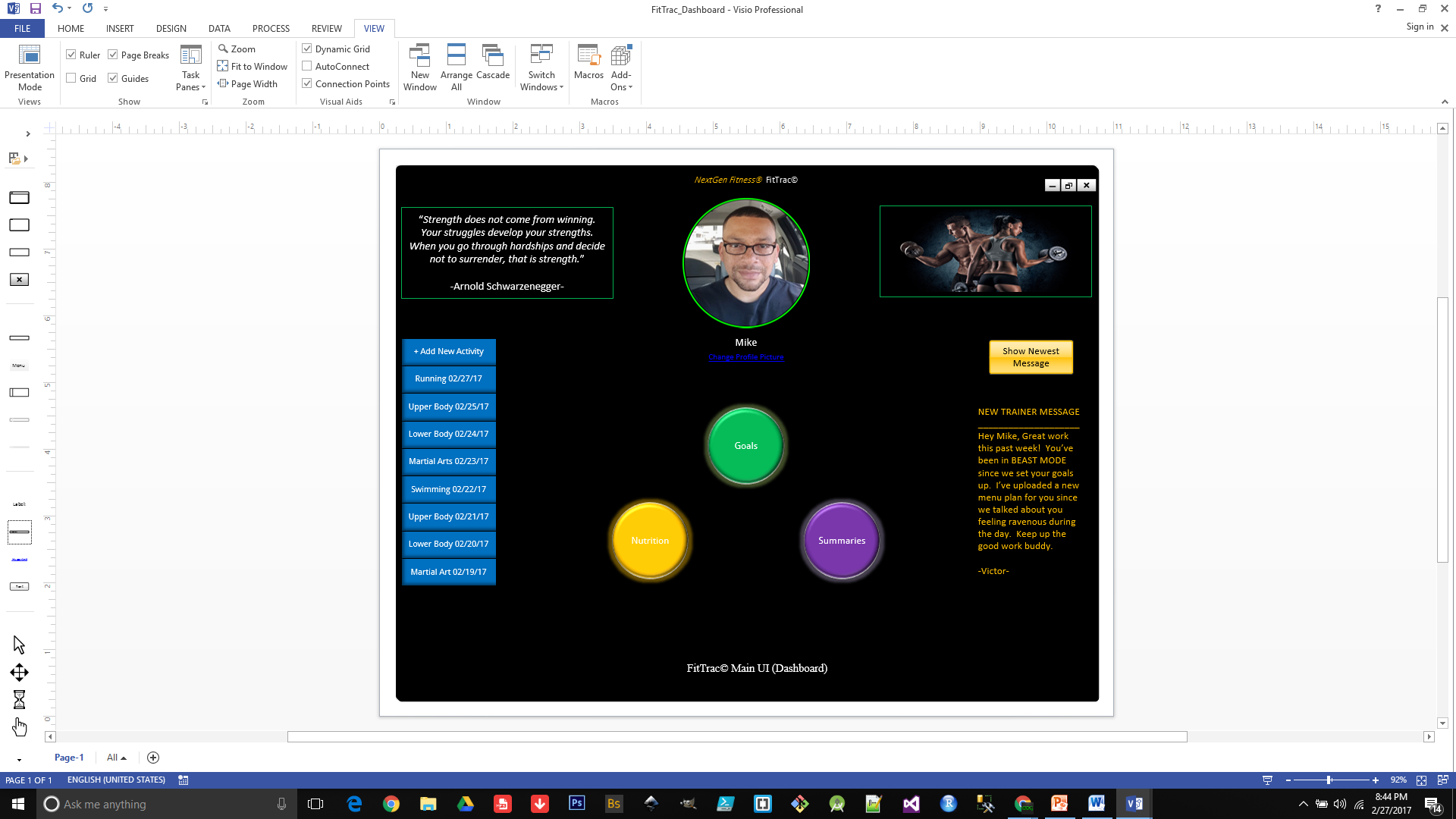
## Sequence Diagram



## High-Level UI (Log-in)



## High-Level UI (Dashboard)



# Week 3: Test and Quality Assurance Plan

## Quality Assurance Plan

### Introduction

This document will serve as the quality assurance (QA) plan for the FitTrac© project. This comprehensive plan will cover the following project areas:

* **QA Team Selection** (Editorial Board, 2016)
* **Scope of QA within each phase** (Editorial Board, 2016)

Meeting timeline and budget constraints will be critical to the success of the software project. In order to ensure that development is conducted in an efficient and effective manner, the QA plan shall be the compass by which all teams will navigate each project phase and define the criteria for meeting “high quality” standards. If for any reason there is a disparity between this document and project requirements, teams shall consult their leads for guidance in all matters of QA. Team leads will report these contradictions to the QA Manager who will notify the management team in order to reconcile the issue(s).

### QA Team Selection

|  |  |
| --- | --- |
| **QA Team Roles** | **QA Team Duties** |
| **QA Manager** | Must hold the position of Project Manager. Overall in charge of the QA team. Reports progress and findings to the management team weekly. Holds final approval / disapproval authority for project phase entrance / exit. |
| **Gold Team Lead** | Must hold the position of Sr. Software Engineer or System Architect. Leads the gold team in the assessment of project compliance with functional requirements. Reports progress and findings to the QA Manager daily. |
| **Gold Team Member 1** | Holds the position of Software Engineer or Developer. Reports progress and findings to Team Lead daily. |
| **Gold Team Member 2** | Holds the position of Software Engineer or Developer. Reports progress and findings to Team Lead daily. |
| **Gold Team Member 3** | Holds the position of Software Tester. Reports progress and findings to Team Lead daily. |
| **Blue Team Lead** | Must hold the position of Sr. Software Engineer or Systems Architect. Leads the blue team in the assessment of project compliance with non-functional requirements. Reports progress and findings to the QA Manager daily. |
| **Blue Team Member 1** | Holds the position of Software Engineer or Developer. Reports progress and findings to Team Lead daily. |
| **Blue Team Member 2** | Holds the position of Software Engineer or Developer. Reports progress and findings to Team Lead daily. |
| **Blue Team Member 3** | Holds the position of Technical Writer. Reports progress and findings to Team Lead daily. |

### Scope

**Planning and Requirements Analysis Phase**

During this phase, QA team members will be selected in accordance with plan criteria. QA methods and controls that will be implemented will be selected for each of the following phases as well as the setting of checkpoints and milestones. QA audits, review, and testing standards will be determined and documented.

**Requirements Definition Phase (Milestone ‘A’)**

**Documentation:** Requirements Traceability Matrix

**Method:** QA audit

QA blue and gold team members shall audit the RTM in accordance with checklists and their area of responsibility. Each checklist shall be serialized with a check point ID number. These audits will ensure that requirements are clear, detailed, and contain testing success criteria. Once an audit is complete, checklists shall be passed to team leads for review. After reviewing the checklists, leads shall pass audit findings along with a copy of the RTM to the QA Manager who will give a final analysis of ‘PASS’ or ‘FAIL’ for the milestone. Audit results shall be reported to the management team and a determination will be made for exit or retention in the current phase.

**RTM Audit Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKA001** | |  | |
| Requirement Statement | Requirements statements are definitive. Statements address specific stakeholder requirements. | |  |  |
| Priority | A priority number is present and is not a duplicate. | |  |  |
| Source | Source is present and conforms to a primary or secondary stakeholder. | |  |  |
| Risk | Risk assessment is clear and unambiguous | |  |  |
| Opportunity | Opportunity assessment is clear and unambiguous. | |  |  |
| Category | Category listed falls under a functional or non-functional requirement. | |  |  |
| Test Acceptance Criteria | Acceptance criteria are specific, measurable, and can be answered with ‘Yes’ or ‘No’. | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Design Phase (Milestone ‘B’)**

**Documentation:** Diagrams

**Method:** QA Audit

QA team members shall audit all project diagrams in accordance with checklists. Each checklist shall be serialized with a check point ID number. These audits will ensure that diagrams are graphically logical, detailed, and address the specific function of the system. Once an audit is complete, checklists shall be passed to team leads for review. After reviewing the checklists, leads shall pass audit findings along with copies of diagrams to the QA Manager who will give a final analysis of ‘PASS’ or ‘FAIL’ for the milestone. Audit results shall be reported to the management team and a determination will be made for exit or retention in the current phase.

**Diagram Audit Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKB001** | |  | |
| Use Case Diagram | Diagram contains appropriate actors and systems. Actor and sub-system interaction are logical and clear. Interactions have call-outs with amplifying information related to their flow of data. | |  |  |
| Functional Decomposition Diagram | Diagram systems and sub-systems are numbered and demonstrate correct parent/ child relationships. | |  |  |
| Data Flow Diagrams | Diagrams are named for their purpose or function. External entities, processes, and data stores are labeled. Data flow between entities, processes, and data stores is logical. Data flow has call-out or amplifying information related to its purpose. | |  |  |
| Sequence Diagram | Diagram is named. Entities are labeled. Events are logical and have call-outs or amplifying information related to their purpose. | |  |  |
| UI Mock-ups |  | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Development and Implementation Phase (Milestone ‘C’)**

**Documentation:** Pseudocode, programming documentation

**Method:** QA Review

QA team members shall conduct a review of all project pseudocode and programming documentation in accordance with checklists. Each checklist shall be serialized with a check point ID number. These reviews will ensure that coding documentation is logical, detailed, and addresses the specific function of the system. Once a review is complete, checklists shall be passed to team leads for review. After reviewing the checklists, leads shall pass review findings along with copies of coding documentation to the QA Manager who will give a final analysis of ‘PASS’ or ‘FAIL’ for the milestone. Review results shall be reported to the management team and a determination will be made for exit or retention in the current phase.

**Code Review Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKC001** | |  | |
| Pseudocode | Pseudocode is clear and contains meaningful comments. Code branching is logical and covers cases for all decisions. Algorithms are correct and logically verifiable. | |  |  |
| Design patterns | Code follows the appropriate design patterns for each class or interface. | |  |  |
| Architecture | Separation of concerns in the code is clear and loosely coupled. | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Testing Phase (Milestone ‘D’)**

**Documentation:** RTM, coding documentation, Data Flow, Sequence, and Functional Decomposition diagrams, Test Case Matrices (Guru99, Copyright 2017), Test plans

**Method:** Testing, QA validation and verification, QA Audit

After testing is complete, QA team members shall perform validation and verification on the project in accordance with checklists. Team members shall also conduct audits on test case matrices and test plans by means of their corresponding checklists. Each checklist shall be serialized with a check point ID number. These audits along with validation and verification will ensure that all RTM requirements were met by each test case and that the software adheres to design plans (Editorial Board, 2016). Once audits, verifications, and validations are complete, checklists shall be passed to team leads for review. After reviewing the checklists, leads shall pass their findings along with copies of test case matrices to the QA Manager who will give a final analysis of ‘PASS’ or ‘FAIL’ for the milestone. Results shall be reported to the management team and a determination will be made for exit or retention in the current phase.

**Unit Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD001** | |  | |
| Inputs | Methods accept data from automated tests | |  |  |
| Transformations | Methods perform calculations as per design | |  |  |
| Outputs | Methods display the correct information as per design | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Functional Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD002** | |  | |
| Goals Functions | Goals can be created, edited, and deleted; Goals manager send updates to summary manager in real time | |  |  |
| Nutrition Functions | Menu plans can be uploaded, exported, viewed, and printed; Member accounts have no access to this function | |  |  |
| Summary Functions | Summaries can be viewed and exported; Vital Stats can be edited and viewed | |  |  |
| Activity Functions | Activities can be added and viewed; Activity wizard is easy to follow and all fields can be updated with the correct data | |  |  |
| Profile Picture Functions | Profile pictures can be uploaded, changed, and deleted | |  |  |
| Log-in Functions | Log-in is easy to understand and functions according to design | |  |  |
| Random Quote Generator | Random quotes are clear and properly formatted. | |  |  |
| Random Picture Generator | Random pictures are not pixelated, are clear, and of correct size | |  |  |
| Account Registration Functions | An account can be created with the appropriate access levels based on access code | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**GUI Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD003** | |  | |
| Functionality | UI elements perform as designed | |  |  |
| Color Scheme | Color scheme fits with requirements | |  |  |
| Readability | Font is consistent with style and requirements | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Security Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD004** | |  | |
| Log-in Functions | Registered accounts can successfully log-in to the system; Disabled accounts or incorrect credentials do not allow access | |  |  |
| Forms Security | Forms do not allow the injection of code | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**Stress Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD005** | |  | |
| High-Load System Behavior | System maintains expected performance levels under max user load | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

**User Acceptance Test Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CHECK POINT FIELDS | DESCRIPTION | | MEETS CRITERIA | DOES NOT MEET CRITERIA |
| ID | **CHKD006** | |  | |
| Function | Software functions according to requirements | |  |  |
| Form | UX experience was good to satisfactory | |  |  |
|  |  | |  |  |
| FINAL ANALYSIS | | | | |
| TOTAL MET | |  | | |
| TOTAL NOT MET | |  | | |
| Milestone PASS / FAIL | |  | | |
|  | | | | |
| NOTES | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

## Unit Test Plan

**Documentation**: Data Flow and Sequence Diagrams, Test Case Matrix (Guru99, Copyright 2017), coding documentation

**Methods**: Class method, automated testing

Software testers shall conduct unit tests on various classes and methods in order to uncover defects that have occurred during the coding stages (Editorial Board, 2016). Testing shall be conducted accordance with test checklists. Once testing results are available, copies of the checklists will be forwarded to QA team leads and project managers for review.

**Unit Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement ID | Test Case | Test Steps | Test Data | Expected Results |
| **UT1** | **FIT002** | Call createGoal() | Run automated test to call method | Sample values assigned to variables via automated test | Values assigned with no errors or anomalies |
| **UT2** | **FIT002** | Call editGoal() | Run automated test to call method | Sample values changed and assigned to variables via automated test | Values changed and assigned with no errors or anomalies |
| **UT3** | **FIT002** | Call deleteGoal() | Run automated test to call method | Sample values assigned via automated test | Variable values deleted |
| **UT4** | **FIT003** | Call addActivity() | Run automated test to call method | Sample values assigned to variable via automated test | Values assigned with no errors or anomalies |
| **UT5** | **FIT003** | Call viewActivity() | Run automated test to call method | Variable assignments via automated test | Sample activities displayed with no errors or anomalies |
| **UT6** | **FIT004** | Call uploadMenu() | Run automated test to call method | Sample plan via automated test | Sample menu uploaded to menu class object with no errors or anomalies |
| **UT7** | **FIT004** | Call viewMenu() | Run automated test to call method | Sample plan via automated test | Sample menu displayed with no errors or anomalies |
| **UT8** | **FIT004** | Call exportMenu() | Run automated test to call method | Sample plan via automated test | Menu exported to device with no errors or anomalies |
| **UT9** | **FIT004** | Call printMenu() | Run automated test to call method | Sample plan via automated test | Print job sent to device, menu printed with no errors or anomalies |
| **UT10** | **FIT005** | Call editStats() | Run automated test to call method | Sample stats assigned via automated test | Values changed and assigned with no errors or anomalies |
| **UT11** | **FIT005** | Call viewStats() | Run automated test to call method | Sample stats assigned via automated test | Stats displayed with no errors or anomalies |
| **UT12** | **FIT005** | Call viewFitSum() | Run automated test to call method | Goal input variables assigned via automated test | Summary displayed with no errors or anomalies |
| **UT13** | **FIT005** | Call exportFitSum() | Run automated test to call method | Sample goal summary via automated test | Summary values exported to device with no errors or anomalies |
| **UT14** | **FIT006** | Call addProfPic() | Run automated test to call method | Sample image assigned via automated test | Image uploaded to Profile Pic class object with no errors or anomalies |
| **UT15** | **FIT006** | Call changeProfPic() | Run automated test to call method | Sample image assigned via automated test | Image changed with no errors or anomalies |
| **UT16** | **FIT006** | Call deleteProfPic() | Run automated test to call method | Sample image assigned via automated test | Image deleted |
| **UT17** | **FIT007** | Call randQuote() | Run automated test to call method | Quotes from quote database | Random quotes displayed until loop exit with no errors or anomalies |
| **UT18** | **FIT008** | Call randFitPic() | Run automated test to call method | Fitness images from image database | Random pictures displayed until loop exit with no errors or anomalies |
| **UT19** | **FIT009** | Call createAcct() | Run automated test to call method | Sample values assigned to variables via automated test | Account created and values assigned with no errors or anomalies |

## System Testing Plan

Documentation: RTM, Data Flow, Sequence, and Functional Decomposition Diagrams, Test Case Matrices (Guru99, Copyright 2017)

Methods: GUI, Functional, Stress and Security testing

Software testers shall test GUI elements to ensure their planned functionality and ease of use for the end-user (Editorial Board, 2016). Functional tests shall correlate to the capabilities of the application as a whole (Editorial Board, 2016). Stress tests shall be conducted to observe that the system can handle the planned maximum load of concurrent users (Guru99, Copyright 2017). Security testing aims to find vulnerabilities in various areas of the software (Guru99, Copyright 2017). Testing shall be conducted accordance with RTM requirements and test checklists. Once testing results are available, copies of the checklists will be forwarded to QA team leads and project managers for review.

**Functional Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement ID | Test Case | Test Steps | Test Data | Expected Results |
| **ST1** | **FIT002** | Create a fitness goal | 1. Click ‘Goals’ button 2. After Goals sub menu buttons appear, click ‘New Goal’ 3. Enter information in the required fields of the goals wizard 4. Click the ‘Complete’ button to save goals information | Goal title = Lose 10lbs in 60 days  Goal type = Weight loss  Goal time = 60 days | Goal successfully created |
| **ST2** | **FIT002** | Edit a fitness goal | 1. Click ‘Goals’ button 2. After Goals sub menu buttons appear, click ‘Edit Goal’ 3. Select goal to be edited – “Lose 10lbs. in 60 days” 4. Changed desired goals information 5. Click ‘Complete’ button to save new goal information | Goal title = Lose 10lbs in 45 days  Goal time = 45 days | Goal successfully edited |
| **ST3** | **FIT002** | Delete a fitness goal | 1. Click ‘Goals’ button 2. After Goals sub menu buttons appear, click ‘Delete Goal’ 3. Select goal to be deleted – “Lose 10lbs. in 45 days” 4. Click ‘Delete’ button 5. Click ‘Yes’ button to confirm that the goal is to be deleted | None | Goal successfully deleted |
| **ST4** | **FIT003** | Add a fitness activity | 1. Click ‘Add Activity’ button 2. Input activity data into correct forms via the activity wizard 3. Click ‘Complete’ button to save new goal information | Activity title = Running  Activity date = 02/02/2017  Activity duration = 60 mins.  Activity distance = 7 miles  Activity notes = “Felt great this time” | Activity successfully created |
| **ST5** | **FIT003** | View a fitness activity | 1. Click on a created activity button in the activities panel | None | Activity information successfully displayed |
| **ST6** | **FIT004** | Upload a menu plan | 1. Log-in to trainer account 2. Select the intended client account for the menu 3. Click upload ‘Upload Plan’ button 4. Navigate to menu plan on device and single click to select 5. Click ‘Okay” button to finalize the upload | Sample menu plan | Menu plan successfully uploaded to client account |
| **ST7** | **FIT004** | View a menu plan | 1. Log-in to client account 2. Click the ‘Nutrition’ button 3. After Nutrition sub menu buttons appear, click ‘View Menus’ button 4. Click the desired menu icon | Sample menu plan | Menu plan successfully displayed |
| **ST8** | **FIT004** | Export a menu plan | 1. Log-in to client account 2. Click the ‘Nutrition’ button 3. After Nutrition sub menu buttons appear, click ‘Export Menus’ button 4. Click the desired menu icon | Sample menu plan | Menu plan successfully exported to device |
| **ST9** | **FIT004** | Print a menu plan | 1. Log-in to client account 2. Click the ‘Nutrition’ button 3. After Nutrition sub menu buttons appear, click ‘Print Menus’ 4. Click the desired menu icon 5. Click the ‘Print’ button 6. Select installed printer 7. Click ‘Okay’ button | Sample Menu Plan | Print job sent to printer, Menu plan successfully printed |
| **ST10** | **FIT005** | Edit vital stats | 1. Log-in to client or member account 2. Click the ‘Summaries’ button 3. After Summaries sub menu buttons appear, click ‘Edit Stats’ button 4. Input stats into correct forms 5. Click ‘Complete’ button to save stats information | Resting Heart Rate = 50 bpm  Body Fat (%) = 15  Total Weight (lbs.) = 210  Basal Metabolic Rate (kcal) = 1780 | Vital Stats successfully edited |
| **ST11** | **FIT005** | View vital stats | 1. Log-in to client or member account 2. Click the ‘Summaries’ button 3. After Summaries sub menu buttons appear, click ‘View Vital Stats’ button | Sample Stats | Vital Stats successfully displayed |
| **ST12** | **FIT005** | View fitness summary | 1. Log-in to client or member account 2. Click the ‘Summaries’ button 3. After Summaries sub menu buttons appear, click ‘View Fitness Summary’ button | Sample summary from goals input | Fitness Summary successfully displayed |
| **ST13** | **FIT005** | Export fitness summary | 1. Log-in to client or member account 2. Click the ‘Summaries’ button 3. After Summaries sub menu buttons appear, click ‘Export Summary’ button | Sample summary from goals input | Fitness Summary successfully exported to device |
| **ST14** | **FIT006** | Upload a profile image | 1. Log-in to client or member account 2. Click the ‘Change Profile Picture’ link 3. When the profile picture menu is displayed, click the ‘Upload Image’ button 4. Navigate to the desired image and click to select 5. Click ‘Okay’ button to finalize the upload | Sample image | Profile Picture successfully uploaded |
| **ST15** | **FIT006** | Change a profile image | 1. Log-in to client or member account 2. Click the ‘Change Profile Picture’ link 3. When the profile picture menu is displayed, click the ‘Change Image’ button 4. Navigate to the desired image and click to select 5. Click ‘Okay’ button to finalize the upload | New sample image | Profile Picture successfully changed to new image |
| **ST16** | **FIT006** | Delete a profile image | 1. Log-in to client or member account 2. Click the ‘Change Profile Picture’ link 3. When the profile picture menu is displayed, click the ‘Delete Image’ button | Sample image | Profile Picture successfully deleted |
| **ST17** | **FIT009** | Create a client account | 1. In web browser, enter www.nextgenfitness.com/fitfitt/login 2. On the Log-in window, click the ‘Register’ link 3. Enter data into required fields 4. Click ‘Okay’ button to finalize account creation 5. Log-in with user name and password for account | First name = Sample  Last name = Client  D.O.B. = 04/19/1994  Address = 12345 Sample Dr.  City = Houston  State = TX  Zip code = 77034  Username = Client001  Password = P@55w0rd!  Client ID = NGFC001 | Client account successfully created |
| **ST18** | **FIT009** | Create a trainer account | 1. In web browser, enter www.nextgenfitness.com/fitfitt/login 2. On the Log-in window, click the ‘Register’ link 3. Enter data into required fields 4. Click ‘Okay’ button to finalize account creation 5. Log-in with user name and password for account | First name = Super  Last name = Trainer  D.O.B. = 11/11/1981  Address = 54321 Sample Blvd  City = Houston  State = TX  Zip code = 77564  Username = Trainer001  Password = P@55w0rd!  Trainer ID = NGFT001 | Trainer account successfully created |
| **ST19** | **FIT009** | Create a member account | 1. In web browser, enter www.nextgenfitness.com/fitfitt/login 2. On the Log-in window, click the ‘Register’ link 3. Enter data into required fields 4. Click ‘Okay’ button to finalize account creation | First name = Justin  Last name = Member  D.O.B. = 06/24/1975  Address = 34521 Sample Ln.  City = Houston  State = TX  Zip code = 77004  Username = Member001  Password = P@55w0rd!  Trainer ID = nil | Member account successfully created |
| **ST20** | **FIT009** | Create a manager account | 1. In web browser, enter www.nextgenfitness.com/fitfitt/login 2. On the Log-in window, click the ‘Register’ link 3. Enter data into required fields 4. Click ‘Okay’ button to finalize account creation | First name = Clint  Last name = Manage  D.O.B. = 06/24/1975  Address = 26512 Sample ST.  City = Houston  State = TX  Zip code = 77193  Username = Manager001  Password = P@55w0rd!  Mgr. ID = NGFM001 | Manager account successfully created |
| **ST21** | **FIT010** | System is Mac and PC compatible | 1. Access system via PC 2. Access system via Mac | Log-in credentials | Successful access to system |

**GUI Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | GUI Element | Test Case | Test Steps | Test Data | Expected Results |
| **GT1** | **Buttons** | Buttons function as per their code design | Click all buttons | None | Buttons function as designed |
| **GT2** | **Drop-down boxes** | Drop-down boxes function as per their code design, Populated with required data, no misspellings | Click all drop-down boxes | Data entered into form | Drop-down boxes function as designed |
| **GT3** | **Links** | Links take user to associated content as per their design | Click all links | None | Links function as designed |
| **GT4** | **Input forms** | Input forms accept user input | Enter correct and incorrect data into all forms | None | Input forms accept correct data and show error upon incorrect data entry |
| **GT5** | **Window Resize** | Window can be resized and content adjusts | Resize Window | None | Content is responsive to window size change and adjusts to fit the window |
| **GT6** | **Colors** | All colors fit with UI/UX design plans | Observe color scheme | None | Color scheme complies with design plans |
| **GT7** | **Fonts** | Character font adhere to UI/UX design plan | Observe character fonts | None | Character fonts comply with design plan |

**Security Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement ID | Test Case | Test Steps | Test Data | Expected Results |
| **SecT1** | **FIT009** | Attempt Log-in to system with incorrect credentials | 1. Enter incorrect username 2. Enter incorrect password 3. Click ‘Enter’ button | User name = BADUSR  Password = sUpErHaCk!!009 | Log-in failed message |
| **SecT2** | **FIT009** | Attempt Log-in to a disabled member account | 1. Disable member account 2. Enter username 3. Enter password 4. Click ‘Enter’ button | User name = Member001  Password = P@55w0rd! | Disabled account message |
| **SecT3** | **FIT009** | Attempt Log-in to a disabled client account | 1. Disable client account 2. Enter username 3. Enter password 4. Click ‘Enter’ button | User name = Client001  Password = P@55w0rd! | Disabled account message |
| **SecT4** | **FIT009** | Attempt Log-in to a disabled trainer account | 1. Disable trainer account 2. Enter username 3. Enter password 4. Click ‘Enter’ button | User name = Trainer001  Password = P@55w0rd! | Disabled account message |
| **SecT5** | **FIT009** | Attempt Log-in to a disabled manager account | 1. Disable manager account 2. Enter username 3. Enter password 4. Click ‘Enter’ button | User name = Manager001  Password = P@55w0rd! | Disabled account message |

**Stress Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement ID | Test Case | Test Steps | Test Data | Expected Results |
| **StrT1** | **FIT001** | Simulate 450 to 500 concurrent users of the system | 1. Integrate load test tool into network 2. Create 450 to 500 users to Log-in to the system | Generated by load test tool | 450 to 500 concurrent IPs active in the system |

## User Acceptance Test Plan

**Documentation:** RTM, Test Case Matrix (Guru99, Copyright 2017)

**Method:** User Acceptance Test

NextGen Fitness management and lead personal training staff shall serve as subject matter experts and participate in acceptance testing (Guru99, Copyright 2017). These tests shall include high-level observation of multiple areas of the software in order to ensure that the product was built to specification and performs as desired (Editorial Board, 2016). Testing shall take place over a span of 30 days within individual fitness clubs. Testing shall be conducted accordance with RTM requirements and test checklists. Once testing results are available, copies of the checklists will be forwarded to QA team leads and project managers for review.

**User Acceptance Test Case Matrix**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Requirement ID | Test Case | Test Steps | Test Data | Expected Results |
|  |  |  | Server Volume |  |  |
| **UAT1** | **FIT001** | Up to 500 concurrent users | 1. Software is installed on NextGen Fitness servers 2. Software is opened for use as a “pilot program” | Member, client, trainer, and manager accounts | System will maintain high performance under user load |
|  |  |  |  |  |  |
|  |  |  | Account Creation and Validation |  |  |
| **UAT2** | **FIT009** | System accessible only to registered users | 1. Delivery of access codes for clients, trainers, and managers 2. User registrations for site access 3. Management audits of client and member accounts to ensure they have the correct access | User created Log-in credentials  Access codes for client, trainer, and management users | System allows the registration of user accounts with the appropriate access levels |
|  |  |  |  |  |  |
|  |  |  | Goals Management |  |  |
| **UAT3** | **FIT002** | Users can create, edit, and delete up to five fitness goals | 1. Users create one to five fitness goals 2. Users edit one to all fitness goals 3. Users delete one to five fitness goals | User created fitness goals | System supports the creation, editing, and deletion of one to five fitness goals |
|  |  |  |  |  |  |
|  |  |  | Activity Management |  |  |
| **UAT4** | **FIT003** | Users can add and view fitness activities | 1. Users create fitness activities 2. Users view their created activities | User created fitness activities | System supports the adding and viewing of fitness activities |
|  |  |  |  |  |  |
|  |  |  | Nutrition Management |  |  |
| **UAT5** | **FIT004, FIT012** | Trainers can upload one to three menu plans; Trainers and clients can view, export, and print one to three menu plans; Managers can view menu plans; Members have no access to menu plans | 1. Trainers create and upload one to three menu plans 2. Trainers view, export, and print one to three menu plans 3. Clients view, export, and print one to three menu plans 4. Managers view menu plans 5. Check that member accounts have menu plan access disabled | Trainer created menu plans | Systems supports user upload, export, view, and print of menu plans depending on account access levels |
|  |  |  |  |  |  |
|  |  |  | Vital Stats Management |  |  |
| **UAT6** | **FIT005** | Users can edit and view their vital stats | 1. Users edit their vital stats 2. Users view their vital stats | Member/ client vital stats | System supports editing and view of member/ client vital stats |
|  |  |  |  |  |  |
|  |  |  | Summary Management |  |  |
| **UAT7** | **FIT005** | Users can view, export, and print fitness summaries | 1. Users view their fitness summaries 2. Users export their fitness summary to their device 3. Users print their fitness summaries via an installed device printer | Member/ client goals | System supports view, export, and print of user fitness summaries |
|  |  |  |  |  |  |
|  |  |  | Profile Picture Management |  |  |
| **UAT8** | **FIT006** | Users can upload, change, and delete their profile picture | 1. Members/ clients upload an image 2. Members/ clients change and image to a different image 3. Members/ clients delete the image | Member/ client images | System supports upload, change, and deletion of a profile picture |
|  |  |  |  |  |  |
|  |  |  | Random Quote Generation |  |  |
| **UAT9** | **FIT007** | System accesses quote database to display random quotes in its UI element | 1. Log-in to an account 2. Observe randomly changing NextGen Fitness UI element | Uploaded quotes to database | System randomly generates quotes in its UI elements |
|  |  |  |  |  |  |
|  |  |  | Random Fitness Picture Generation |  |  |
| **UAT10** | **FIT008** | System accesses fitness picture database to display random NextGen Fitness related pictures in its UI element | 1. Log-in to an account 2. Observe randomly changing NextGen Fitness UI element | Uploaded pictures to database | System randomly generates pictures in its UI elements |
|  |  |  |  |  |  |
|  |  |  | Mac and PC Support |  |  |
| **UAT11** | **FIT10** | System can be accessed by Mac and PC users | 1. Users access their accounts via Mac system 2. Users access their account via PC system | User systems | System supports Mac and PC systems |

# Week 4: Development Strategy

## Development Plan Outline

Software development proposals outlining outsourcing, insourcing, and hybrid options will allow the company to analyze areas such as total cost, benefits vs. risks, and quality in order to select the most appropriate strategy for the production of the FitTrac© system. Each area of concern will be analyzed and scored to determine if the entire system or one-to-many of its components are candidates for the proposed options (Niccolls, 2016).

### Outsourcing Proposal

1. **Realization** (Niccolls, 2016)
   1. Analysis of the company’s previous experience in outsourcing programs
      1. The company currently has little to no experience in outsourcing.
2. **Goal Setting** (Niccolls, 2016)
   1. What will the company hope to accomplish through outsourcing?
      1. Overcome manning and skill constraints associated with developing large scale, enterprise, 3-tier web applications
         1. Requirement ID: FIT011
      2. Remain under the $800,000 budget cap
         1. Requirement ID: FIT014
      3. Deliver the application to NextGen Fitness before the 18 month deadline.
         1. Requirement ID: FIT013
3. **Participation** (Niccolls, 2016)
   1. What areas of expertise will be required to effectively analyze outsourcing as an option?
      1. NextGen Fitness Executives
         1. Will provide inputs on their long and short term business goals for the application
      2. Company Project Managers
         1. Will provide inputs on the company’s business strengths and weaknesses in areas such as staffing, scheduling, and compliance issues
      3. Sr. Software Engineers / System Architects
         1. Will provide inputs on strengths and weaknesses in the company’s development teams, technologies, and application architectural needs
      4. Sales and Marketing Managers
         1. Will provide inputs on resource procurement
      5. Accounting Managers
         1. Will provide inputs on typical estimated costs and budgeting for outsourcing
4. **Identification of Impact Areas** (Niccolls, 2016)
   1. Previous Decisions
      1. Since the company has no previous experience in outsourcing, knowledge of best practices, typical associated costs, policies, and regulatory issues will need to be researched
         1. Estimated time for research
            1. 90 to 180 days
   2. Expertise
      1. The company has highly experienced and skilled managers and senior developers, but lacks high skill levels in UI /UX development
         1. Estimated time to fill the skill gap
            1. 90 to 180 days
   3. Quality
      1. Due to lack of experience and skill in UI / UX development, the quality of the application could be negatively impacted
   4. Costs
      1. Detailed within each outsourcing option
   5. Scale
      1. FitTrac© will be a large enterprise application that requires scalability for future growth
      2. The application requires a separation of concerns that will be addressed through the *MVC* (*Model – View – Controller*) design pattern (Tutorialspoint, Copyright 2017)
   6. Security
      1. NextGen Fitness end-users require a secure application to protect their personal information
      2. The application’s data visualization engines are trade secrets. Protecting code assets is highly desirable.
5. **Outsourcing Options** 
   1. Offshoring
      1. Captive Center
         1. Action
            1. Company would open an office in a target country overseas and hire local talent as full-time employees (Editorial Board, 2016)
         2. Costs
            1. Example - Pitampura, India

Compensation (Estimated for 18 months of initial development) : $7,500 \* 10 developers = $75,000US (Editorial Board, 2016)

Office costs (Estimated for 18 months of initial development): $13, 553.50US (Oberoi, 2015)

**Total estimated development costs = $88,553.50US**

Ongoing application support and maintenance

If captive center were to remain open after application release

Compensation (Estimated for 12 months): $5634 \* 10 employees = $56,340US (Editorial Board, 2016)

Office costs (Estimated for 12 months): $9,035.67US (Oberoi, 2015)

**Total estimated ongoing costs = $65,375.67US**

* + - 1. Benefits
         1. Boosts in production and a decrease in labor expenses by bringing low-cost and well-educated locals onboard (Editorial Board, 2016)
         2. The company would maintain complete control of the application’s development and security (Editorial Board, 2016)
         3. Fewer IT, network access, and communications challenges because the center is part of the company (Editorial Board, 2016)
      2. Risks
         1. Larger time zone differences could affect communications between captive center employees and the parent company (Editorial Board, 2016)
         2. The costs associated with opening a captive center in another country could clash with budget constraints (Editorial Board, 2016)
         3. Regulations and compliance policies for opening a captive center in India could be a barrier
    1. Farshoring
       1. Action
          1. Company would hire a third-party development team in a target foreign country (Editorial Board, 2016)
       2. Costs
          1. Example – India

Compensation (Estimated for 18 months of initial development): $7,500 \* 10 developers = $75,000US (Editorial Board, 2016)

**Total Compensation: $75,000US**

Ongoing application support and maintenance

If a development team were contracted in India

Compensation ((Estimated for 12 months): $5,634 \* 10 employees = $56,340US (Editorial Board, 2016)

**Total estimated ongoing costs = $56,340US**

* + - 1. Benefits
         1. This would be the most cost effective offshore solution (Editorial Board, 2016)
      2. Risks
         1. Larger time zone differences could affect communications between farshore employees and the company (Editorial Board, 2016)
         2. Attrition of contract employees could compromise the company’s competitive edge should they take their knowledge to a competitor (Editorial Board, 2016)
    1. Nearshoring
       1. Action
          1. Company would hire a development team in a target neighboring country (Editorial Board, 2016)
       2. Costs
          1. Example – Brazil

Compensation (Estimated for 18 months of initial development): $78,750 \*10 developers = $787,500US (Clifford, 2015)

**Total Compensation: $787,500US**

Ongoing application support and maintenance

If a development team were contracted in Brazil

Compensation (Estimated for 12 months): $52,000 \* 10 employees = $525,000US (Clifford, 2015)

**Total estimated ongoing costs = $525,000US**

* + - 1. Benefits
         1. Reduced production costs (Editorial Board, 2016)
         2. Fewer time zones issues (Editorial Board, 2016)
         3. Lower company travel costs (Editorial Board, 2016)
      2. Risks
         1. Attrition of contract employees could compromise the company’s competitive edge should they take their knowledge to a competitor (Editorial Board, 2016)
  1. Inshoring
     1. Action
        1. The company would hire a local vendor for development (Editorial Board, 2016)
     2. Costs
        1. Example – “Moderate” Class Company (Halyna, 2016)
           1. Project cost range: $15,000 - $50,000
           2. **Total Compensation (avg.) = $35,000US**
        2. Ongoing support and maintenance
           1. Cost range: $15,000 - $50,000
           2. **Total Compensation (avg.) = $35,000US**
     3. Benefits
        1. Getting highly skilled developers who have gone through background checks (Editorial Board, 2016)
        2. No overhead costs from employee benefits (Editorial Board, 2016)
     4. Risks
        1. The more senior the development team, the higher their salary costs (Editorial Board, 2016)

1. **Prioritization** (Niccolls, 2016)
   1. Scoring Metrics
      1. Areas will be graded on scale of 1 – 6 (1 – *Highest Priority*, 6 – *Lowest Priority*)
         1. Quality
         2. Security
         3. Cost
         4. Expertise
         5. Scale
         6. Previous Decisions
2. **Final Analysis**
   1. Because quality, security, and cost are ranked as the projects highest priorities, ***Inshoring*** would provide the potential for the best quality, address security issues through contractual agreements, and keep costs at a moderate level.

### Insourcing Proposal

1. **Realization** (Niccolls, 2016)
   1. Analysis of the company’s previous experience with in-house development
      1. The company currently has a great deal of knowledge related to insource development.
2. **Goal Setting** (Niccolls, 2016)
   1. What will the company hope to accomplish through insourcing?
      1. Maintain high quality levels for the FitTrac system
      2. Maintain data visualization process trade secrets
3. **Participation** (Niccolls, 2016)
   1. What areas of expertise will be required to effectively analyze insourcing as an option?
      1. NextGen Fitness Executives
         1. Will provide inputs on their long and short term business goals for the application
      2. Company Project Managers
         1. Will provide inputs on the company’s business strengths and weaknesses in areas such as staffing, scheduling, and compliance issues
      3. Sr. Software Engineers / System Architects
         1. Will provide inputs on strengths and weaknesses in the company’s development teams, technologies, and application architectural needs
      4. Sales and Marketing Managers
         1. Will provide inputs on resource procurement
      5. Accounting Managers
         1. Will provide inputs on typical estimated costs and budgeting for insourcing
4. **Identification of Impact Areas** (Niccolls, 2016)
   1. Previous Decisions
      1. Since the company has previous experience in insourcing, knowledge of best practices, typical associated costs, policies, and regulatory issues are part of the current knowledge information system.
   2. Expertise
      1. The company has highly experienced and skilled managers and senior developers, but lacks high skill levels in UI /UX development
         1. Estimated time to fill the skill gap
            1. 90 to 180 days
   3. Quality
      1. Due to lack of experience and skill in UI / UX development, the quality of the application could be negatively impacted
   4. Costs
      1. Detailed within insourcing option cost analysis
   5. Scale
      1. FitTrac© will be a large enterprise application that requires scalability for future growth
      2. The application requires a separation of concerns that will be addressed through the *MVC* (*Model – View – Controller*) design pattern (Tutorialspoint, Copyright 2017)
   6. Security
      1. NextGen Fitness end-users require a secure application to protect their personal information
      2. The application’s data visualization engines are trade secrets. Protecting code assets is highly desirable.
5. **Insourcing Analysis** 
   * 1. Action
        1. Company would develop the project in-house (Editorial Board, 2016)
        2. Costs
           1. Compensation (Estimated for 18 months of initial development) : $127,500 \* 10 developers = $850,000US (Editorial Board, 2016)
           2. Tools: Open-source - $0US

**Total estimated development costs = $1,275,000US**

Ongoing application support and maintenance

If the company were to maintain the application

Compensation (Estimated for 12 months): $85,000 \* 10 employees = $850,000US (Editorial Board, 2016)

**Total estimated ongoing costs = $850,000US**

* + - 1. Benefits
         1. Can more easily meet business requirements (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
         2. Will minimize changes in business procedures and policies (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
         3. Will meet constraints of existing systems
         4. Will meet constraints of current technology (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
         5. Will develop new internal resources and capabilities (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
      2. Risks
         1. Higher development and maintenance costs (Editorial Board, 2016)
         2. Loss of time to acquire the needed employees and/ or technology that may be needed for development and maintenance.

1. **Prioritization** (Niccolls, 2016)
   1. Scoring Metrics
      1. Areas will be graded on scale of 1 – 6 (1 – *Highest Priority*, 6 – *Lowest Priority*)
         1. Quality
         2. Security
         3. Cost
         4. Expertise
         5. Scale
         6. Previous Decisions
2. **Final Analysis**
   1. Because quality, security, and cost are ranked as the projects highest priorities, ***Insourcing*** the entire application would not be a cost effective measure. Quality and security for the business and data layers would be maintained.

### Combination Proposal

1. **Realization** (Niccolls, 2016)
   1. Analysis of the company’s previous experience with a “make part / buy part” development process
      1. The company currently has no experience related to a combination development process.
2. **Goal Setting** (Niccolls, 2016)
   1. What will the company hope to accomplish through combination development?
      1. Maintain high quality levels for the FitTrac system
      2. Maintain data visualization process trade secrets
      3. Outsource application components that are cheaper to development and maintain
3. **Participation** (Niccolls, 2016)
   1. What areas of expertise will be required to effectively analyze combination development as an option?
      1. NextGen Fitness Executives
         1. Will provide inputs on their long and short term business goals for the application
      2. Company Project Managers
         1. Will provide inputs on the company’s business strengths and weaknesses in areas such as staffing, scheduling, and compliance issues
      3. Sr. Software Engineers / System Architects
         1. Will provide inputs on strengths and weaknesses in the company’s development teams, technologies, and application architectural needs
      4. Sales and Marketing Managers
         1. Will provide inputs on resource procurement
      5. Accounting Managers
         1. Will provide inputs on typical estimated costs and budgeting for combination development
4. **Identification of Impact Areas** (Niccolls, 2016)
   1. Previous Decisions
      1. Since the company has no previous experience in combination development, knowledge of best practices, typical associated costs, policies, and regulatory issues would require research
         1. Estimated time for research
            1. 90 days
   2. Expertise
      1. The company has highly experienced and skilled managers and senior developers, but lacks high skill levels in UI /UX development
         1. Estimated time to fill the skill gap
            1. 90 to 180 days
   3. Quality
      1. Due to lack of experience and skill in UI / UX development, the quality of the application could be negatively impacted
   4. Costs
      1. Detailed within combination development option cost analysis
   5. Scale
      1. FitTrac© will be a large enterprise application that requires scalability for future growth
      2. The application requires a separation of concerns that will be addressed through the *MVC* (*Model – View – Controller*) design pattern (Tutorialspoint, Copyright 2017)
   6. Security
      1. NextGen Fitness end-users require a secure application to protect their personal information
      2. The application’s data visualization engines are trade secrets. Protecting code assets is highly desirable.
5. **Combination Options**
   1. Commercial Off-The-Shelf (COTS)
      1. Action
         1. Company would purchase part of the application from another company (Editorial Board, 2016)
            1. Costs

Best outsourcing option – Inshoring

Example – “Moderate” Class Company (Halyna, 2016)

Project cost range: $15,000 - $50,000

* + - * 1. **Total Compensation (avg.) = $20,000US**
        2. Ongoing support and maintenance

If the outside company were to maintain part of the application

Example – “Moderate” Class Company (Halyna, 2016)

Project cost range: $15,000 - $50,000

**Total Compensation (avg.) = $20,000US**

* + 1. Action
       1. Company would develop the business and data layers in-house
          1. Compensation (Estimated for 18 months of initial development) : $127,500 \* 4 developers = $510,000US (Editorial Board, 2016)
          2. Tools: Open-source - $0US

**Total estimated development costs = $510,000US**

Ongoing application support and maintenance

If the company were to maintain the business and data layers

Compensation (Estimated for 12 months): $85,000 \* 3 employees = $255,000US (Editorial Board, 2016)

**Total estimated ongoing costs = $255,000US**

* + 1. Benefits
       1. Lower costs (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
       2. Shorter implementation time (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
       3. Require less technically skilled staff (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
       4. Future updates are provided by the vendor (Crispin, IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4), 2017)
    2. Risks
       1. Limited custom functionality (Editorial Board, 2016)
  1. Modified Off-The-Shelf (MOTS)
     1. Action
        1. Company would purchase part of the application with customized features and consume vendor API(s) (Editorial Board, 2016)
           1. Costs

Best outsourcing option – Inshoring

Example – “Moderate” Class Company (Halyna, 2016)

Project cost range: $15,000 - $50,000

* + - * 1. **Total Compensation (avg.) = $40,000US (customized option)**
        2. Ongoing support and maintenance

If the outside company were to maintain part of the application

Example – “Moderate” Class Company (Halyna, 2016)

Project cost range: $15,000 - $50,000

**Total Compensation (avg.) = $40,000US (customized option)**

* + 1. Action
       1. Company would develop the business and data layers in-house
          1. Compensation (Estimated for 18 months of initial development) : $127,500 \* 3 developers = $381,000US (Editorial Board, 2016)
          2. Tools: Open-source - $0US

**Total estimated development costs = $381,000US**

Ongoing application support and maintenance

If the company were to maintain the business and data layers

Compensation (Estimated for 12 months): $85,000 \* 3 employees = $255,000US (Editorial Board, 2016)

**Total estimated ongoing costs = $255,000US**

* + 1. Benefits
       - 1. Lower costs (Editorial Board, 2016)
         2. Require less technically skilled staff (Editorial Board, 2016)
         3. Negotiable terms for vendor modifications and bug fixes (Editorial Board, 2016)
    2. Risks
       - 1. Buying company will have to maintain the application unless terms are negotiated (Editorial Board, 2016)
         2. Buying company will be at the mercy of the vendor who will control pricing and the update release schedule (Editorial Board, 2016).

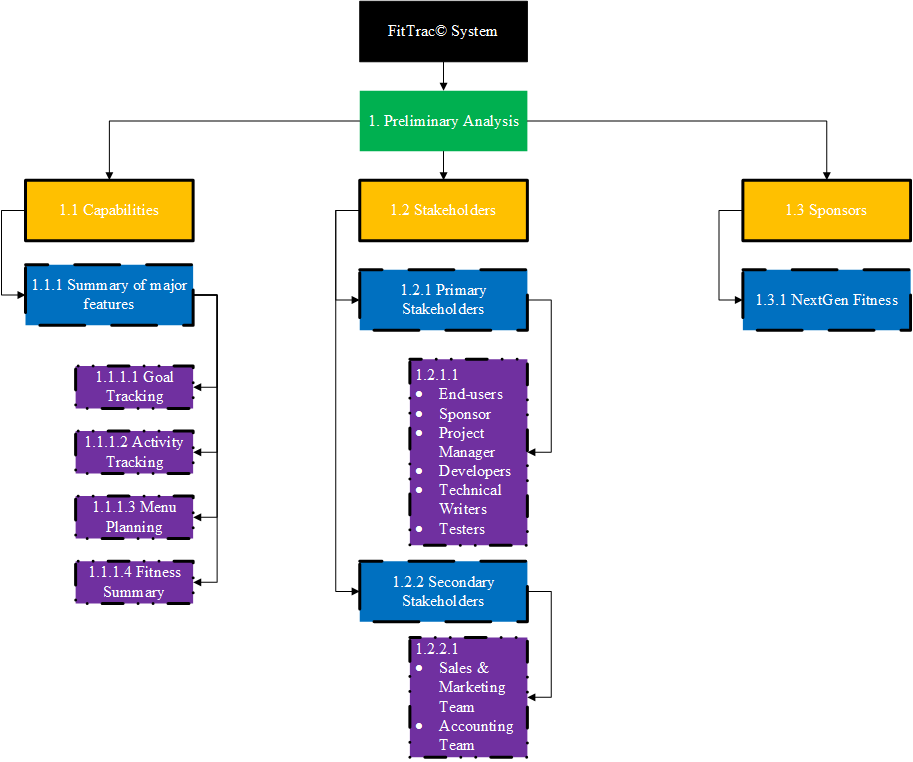
1. **Prioritization** (Niccolls, 2016)
   1. Scoring Metrics
      1. Areas will be graded on scale of 1 – 6 (1 – *Highest Priority*, 6 – *Lowest Priority*)
         1. Quality
         2. Security
         3. Cost
         4. Expertise
         5. Scale
         6. Previous Decisions
2. **Final Analysis**
   1. Because quality, security, and cost are ranked as the projects highest priorities, ***combination development*** would allow a customized view to be developed by a vendor while the company maintains control of the controller and model layers. Changes within the view will occur less frequently than that of the controller and the model so updates and bugs fixes will not drastically impact application functionality.

### Development Strategy Recommendation

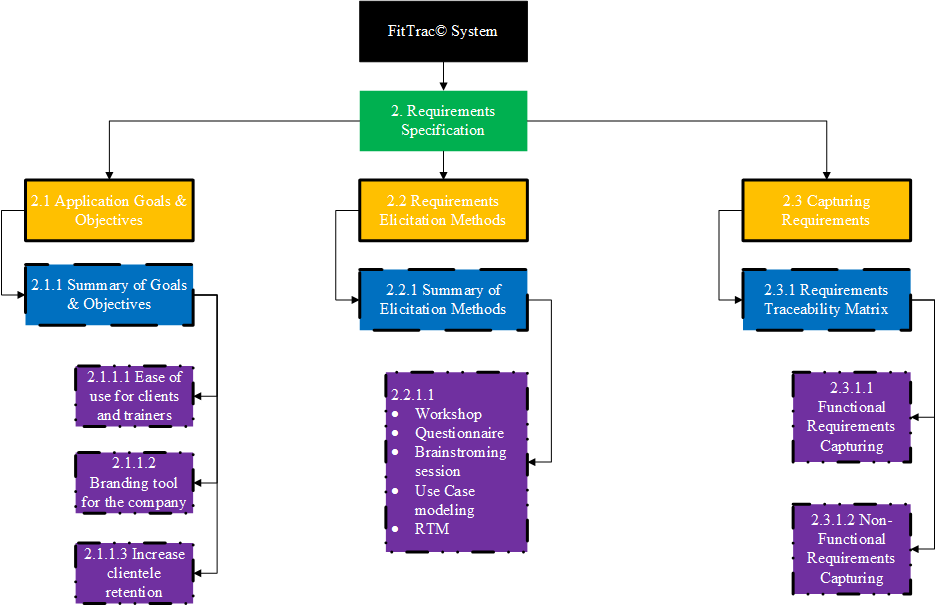
After a complete analysis of the three development strategies, it has been determined that a combination process will meet company and client needs in the short and long term. The hiring of a vendor to perform development of the GUI inshore, although more costly than overseas options, would solve foreign country regulatory, attrition, compensation, and communication issues associated with other processes. Vendor maintenance of the view layer would also solve the skill gap issue that the company is currently facing. Utilizing a local vendor and in-house developers will keep costs well under budget and position the company as the preferred source for future application upgrades and positively impact its reputation. Although updates and bug fixes for the GUI would be driven by the vendor, contractual terms may be created in order to have a predictable schedule and associated billing rate.

# Week 5: Integration and Development Plan

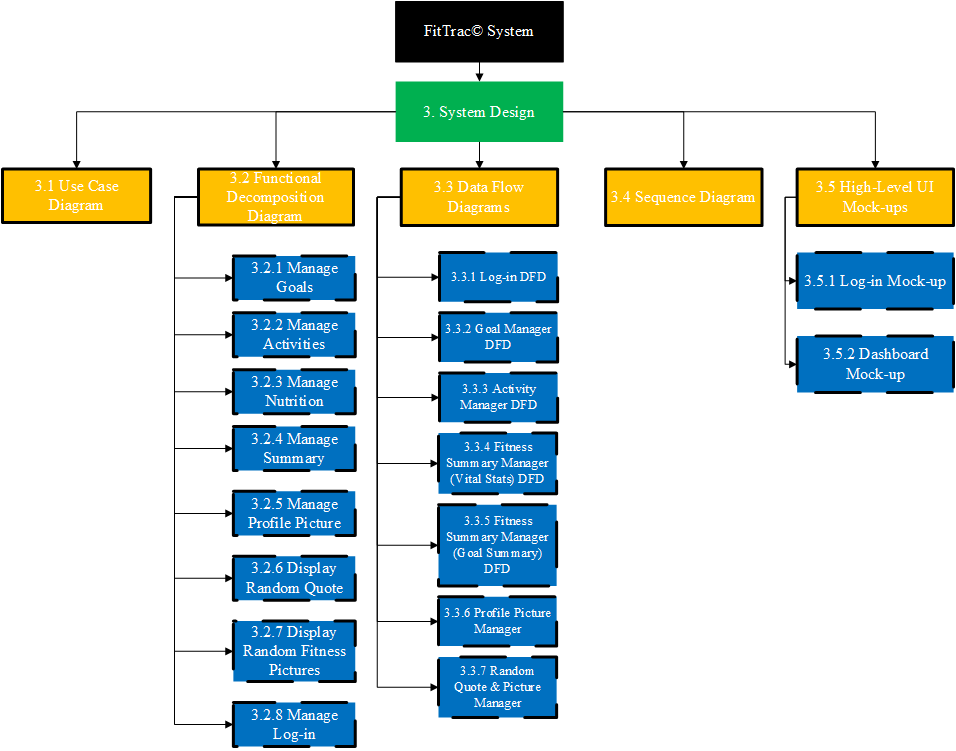
## Work Breakdown Structure – Preliminary Analysis



## Work Breakdown Structure – Requirements Definition



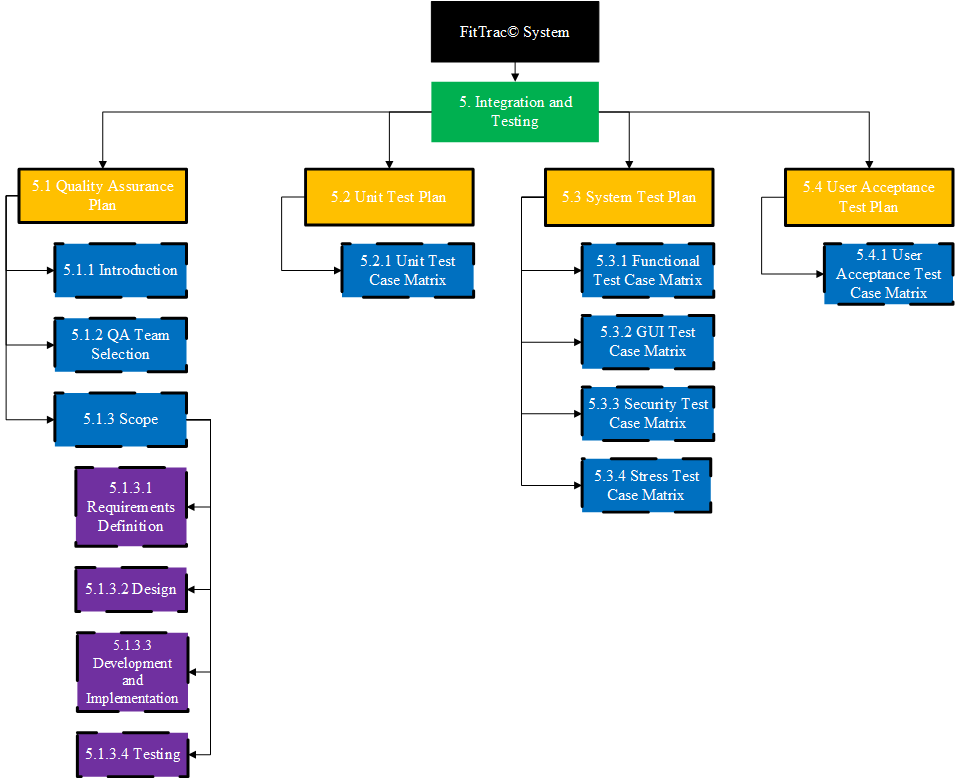
## Work Breakdown Structure – System Design



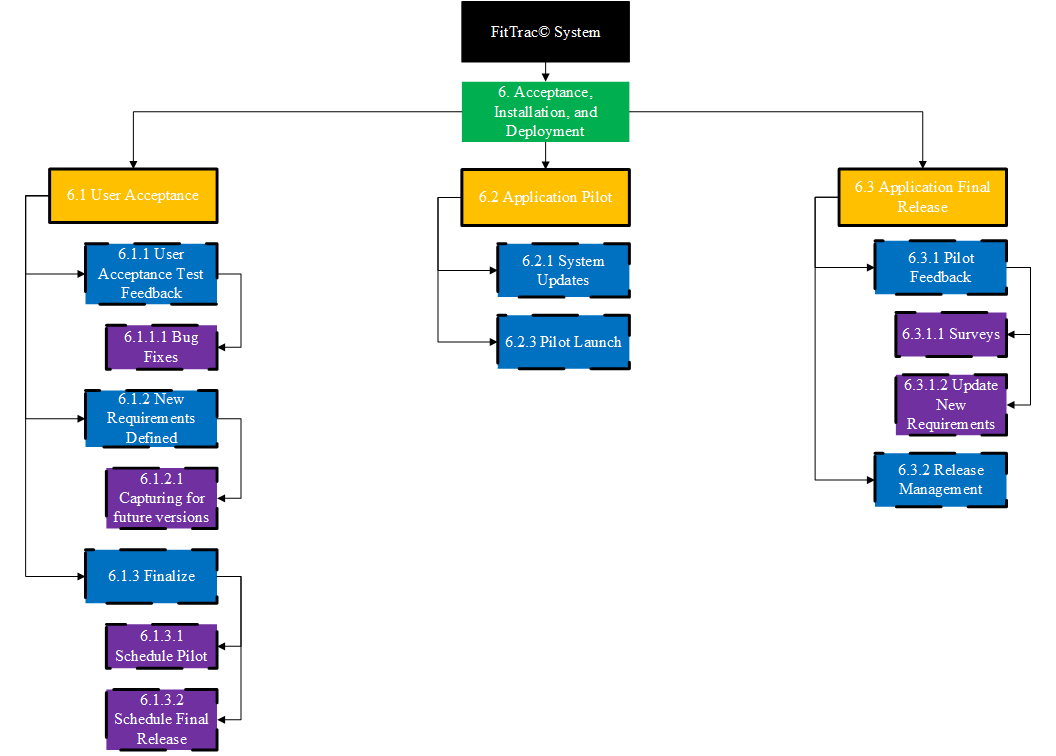
## Work Breakdown Structure – Development



## Work Breakdown Structure – Integration and Testing

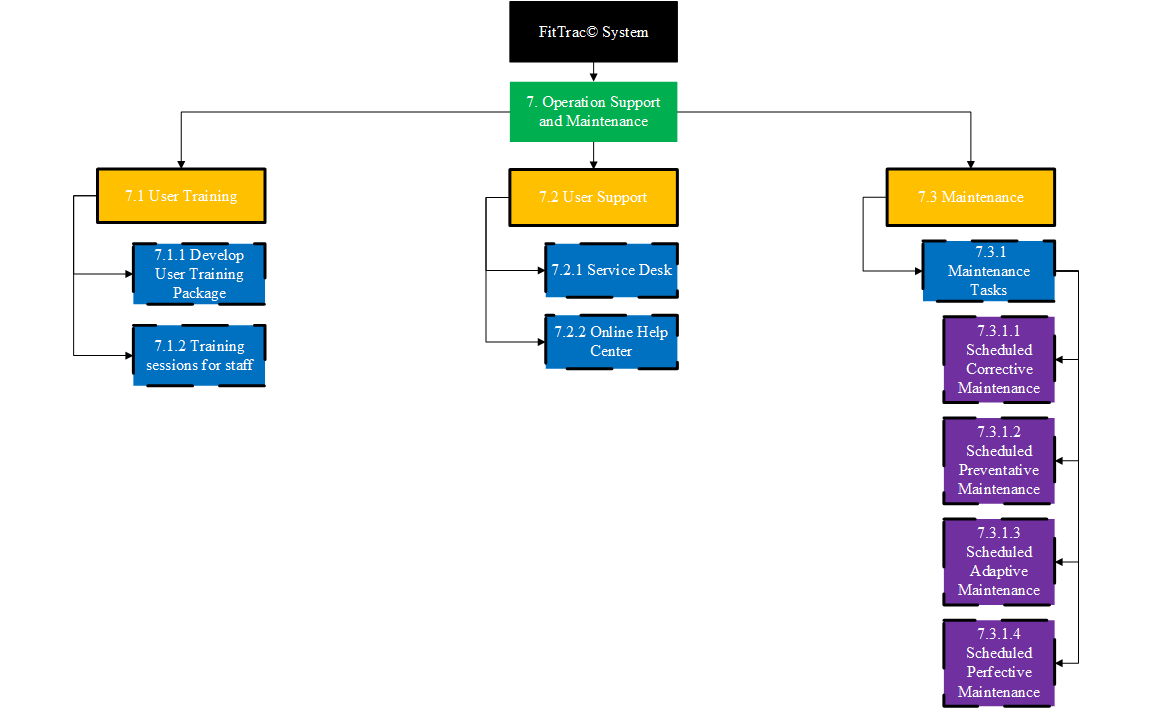


## Work Breakdown Structure – Acceptance, Installation, and Deployment



(Crispin, IT425 - System Analysis, Design, and Integration (PPT Phase 5), 2017)

## Work Breakdown Structure – Operation Support and Maintenance



(Crispin, IT425 - System Analysis, Design, and Integration (PPT Phase 5), 2017)

# References

BrainyQuote. (Copyright 2001 - 2017). *Strength Quotes*. Retrieved February 27, 2017, from Brainy Quote website: https://www.brainyquote.com/quotes/quotes/a/arnoldschw116694.html?src=t\_strength

Clifford, C. (2015, February 5). *Need a Software Engineer? Here's How Much You Can Expect to Pay. (Infographic)*. Retrieved March 11, 2017, from Entrepreneur website: https://www.entrepreneur.com/article/242613

Crispin, J. (2017, March 14). IT425 - System Analysis, Design, and Integration (PPT Phase 5). Colorado Springs, CO, USA. Retrieved March 19, 2017

Crispin, J. (2017, March 7). IT425 - Systems Analysis, Design, and Integration (PPT - Phase 3 & 4). Colorado Springs, CO, USA. Retrieved March 10, 2017

Editorial Board. (2016). Systems Analysis and Design. (1). Shaumburg, IL, USA: Words of Wisdom, LLC. Retrieved February 15, 2017

Guru99. (Copyright 2017). *How to Create Requirements Traceabiliity Matrix (RTM)*. Retrieved March 5, 2017, from Guru99 website: http://www.guru99.com/traceability-matrix.html

Halyna. (2016, July 26). *How Much Does it Cost to Develop a Web App?* Retrieved March 12, 2016, from Romexsoft website: https://www.romexsoft.com/blog/develop-web-app/

Haughey, D. (2014, February 9). *Work Breakdown Structure 101 - The Foundation of Project Planning*. Retrieved March 13, 2017, from Project Smart website: https://www.projectsmart.co.uk/work-breakdown-structure-101.php

Niccolls, C. (2016, August 6). *How to Develop a Plan for Outsourcing*. Retrieved March 11, 2017, from The Balance website: https://www.thebalance.com/how-to-develop-a-plan-for-outsourcing-2553036

Oberoi, A. (2015, January 21). *What and How Much it Takes to Build an Office in the Indian Capital? The AdPushup Story*. Retrieved March 11, 2017, from ADPUSHUP BLOG website: https://www.adpushup.com/blog/new-adpushup-office-features-and-cost-breakdown/

Tutorialspoint. (Copyright 2017). *Design Patterns - MVC Pattern*. Retrieved March 11, 2017, from Tutorialspoint website: https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm

Unknown. (Copyright 2015). *Photo Gallery of Dream - Fitness [image]*. Retrieved February 27, 2017, from We Know Your Dreams website: http://weknowyourdreams.com/fitness.html