

ISEN1000 Introduction to Software Engineering

Curtin College-Bentley

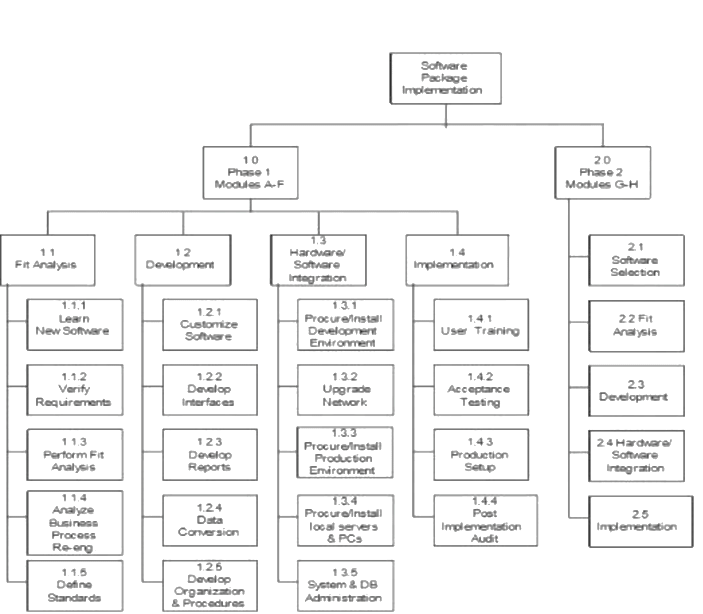


Sumit Khanwani

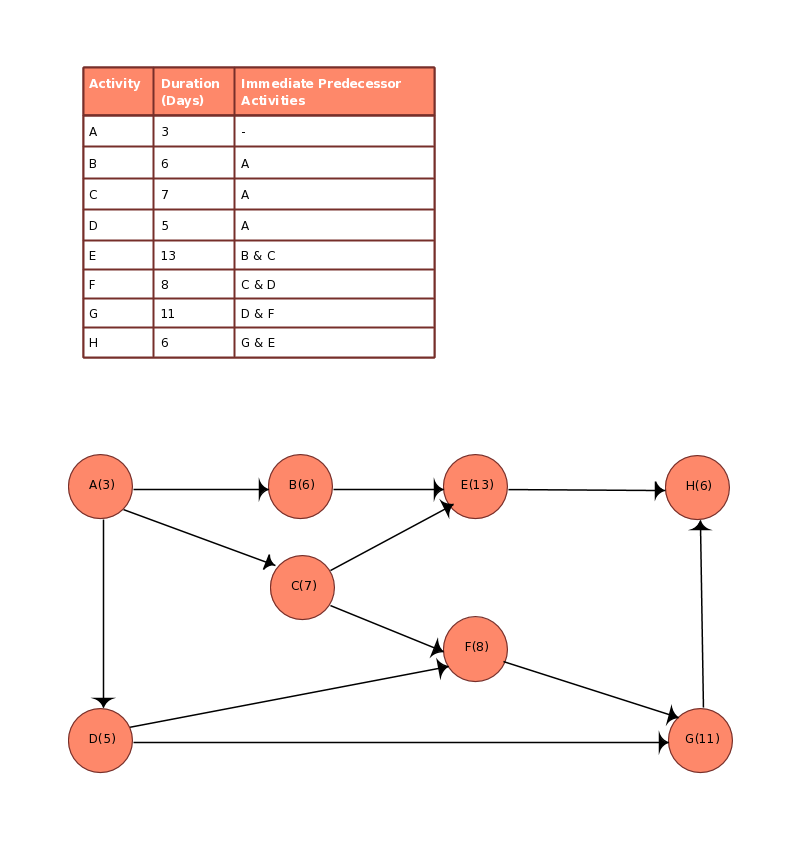
90027977

# **PLANNING:**

## **Work Break Down Structure (WBS)**



## **AON Graph:**



By the help of the diagram, we take out he total number of days required will be 59 days hence the critical path will be A->C->F->G

# **FUNCTIONAL REQUIREMENT ANALYSIS:**

## **ACTORS:**

**Administrator** – In charge of managing extras, subscriptions, goals, and customer accounts.

**Customer** - Records activities, Selects goals, Manages subscriptions, and Views fitness history via mobile app.

**FitTrack Smart Watch** - Routinely sending data to the system.

**Notification Service** - Sends notifications to customers' phones frequently.

**Graphing Service** – Plots graphs and generates fitness & health stats for the app.

# **USER STORIES:**

## **Customer**

As a customer, it should allow me to keep track of my fitness regime via mobile app.

As a customer, it should allow me to set and adjust my fitness goals.

As a customer, it should allow me to opt for extras, such as diet plans and training sessions.

As a customer, it should allow me to check my fitness routine & progress towards my goals.

As a customer, it should allow me to display graphs with colors that are color-blind friendly and high contrast.

## **Administrator**

As an administrator, it should allow me to oversee goals, extras, and subscriptions.

As an administrator, it should allow me to frequently gather & store data from the FitTrack smart watch.

As an administrator, it should allow me to cancel subscriptions.

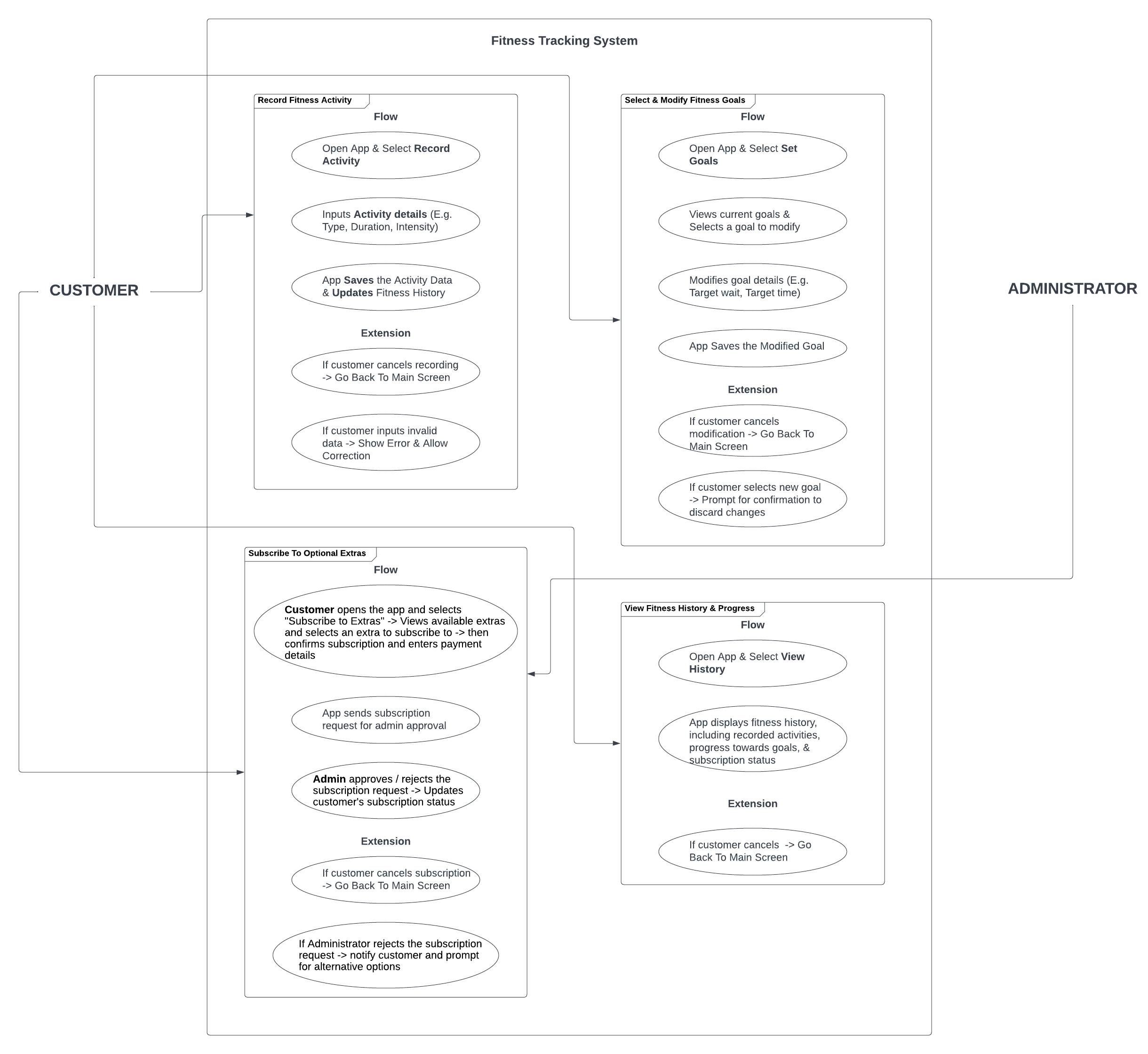
## **Notification Service**

As a notification service, it should allow me to send regular notifications to customers' phones with feedback and recommendations.

## **Graphing Service**

As a graphing service, it should allow me generate fitness and health stats & plot graphs for the app.

# **USE CASES:**



## **Written use case(s), including FOE and extensions**

**Record Fitness Activity:**

**Actors -> Customer**

**Flow**

* Opens the app and selects "Record Activity"
* Inputs activity details (e.g. type, duration, intensity)
* App saves the activity data and updates fitness history

**Extensions**

If customer cancels recording -> go back to main screen

If customer inputs invalid data -> show error message and allow correction

**Select and Modify Fitness Goals:**

**Actors -> Customer**

**Flow**

* Opens the app and selects "Set Goals"
* Views current goals and selects a goal to modify
* Modifies the goal details (e.g. target weight, target time)
* App saves the modified goal

**Extensions**

If customer cancels modification -> go back to main screen

If customer selects a new goal -> prompt for confirmation to discard changes

**Subscribe to Optional Extras:**

**Actors -> Customer, Administrator**

**Flow**

**Customer** opens the app and selects "Subscribe to Extras" -> Views available extras and selects an extra to subscribe to -> then confirms subscription and enters payment details

App sends subscription request to Administrator for approval

**Administrator** approves or rejects the subscription request -> Updates customer's subscription status

**Extensions**

If customer cancels subscription -> go back to main screen

If Administrator rejects the subscription request -> notify customer and prompt for alternative options

**View Fitness History and Progress:**

**Actors -> Customer**

**Flow**

Opens the app and selects "View History"

App displays fitness history, including recorded activities, progress towards goals, and subscription status

**Extensions**

If customer cancels -> go back to main screen

# **NON-FUNCTIONAL REQUIREMENT ANALYSIS:**

## **USABILITY REQUIREMENTS**

The mobile app should have an intuitive and user-friendly interface to record fitness activities, set goals, and regulate subscriptions.

## **PERFORMANCE REQUIREMENTS**

The mobile app should be feasible with any of the mobile operating systems and should support all types of smart watches.

## **RELIABILITY REQUIREMENTS**

The mobile app should be having high security in order to achieve the integrity between the customer and administrator so that the personal data could not be leaked.

# **Version Control**

## **Actual Use of Version Control:**

Version control is used to manage changes to software code, including non-functional requirements (NFRs), over time. It enables developers to track their changes, collaborate with other developers, and maintain code quality and stability throughout the software development lifecycle. With version control, teams can work on different features simultaneously without overwriting each other's work. The version control system maintains a history of changes, which can be used to track progress and identify issues. Overall, version control is critical for efficient, collaborative, and high-quality software development.

## **Plan for Using Version Control:**

To use version control effectively, it's essential to choose the right system for your needs, such as Git or SVN. You need to establish a process for code review, branching, merging, and conflict resolution. You should also define guidelines for committing changes, including commit messages and file naming conventions. It's crucial to train developers on how to use version control properly and ensure they follow best practices. Finally, you should regularly back up your version control system to prevent data loss. By following these guidelines, you can ensure a smooth and efficient development process with version control.