

Toddler Force - Scope Statement

Introduction

Team Name

During our first team meeting we found out that we all have children that are two or three years old. Toddler Force Software was born.

Application Rankings

The following applications were considered as possible projects:

1. Health Tracker - standalone GUI application to track calories against a goal for weight management.
2. Study Buddy - AI-assisted method to enter course materials and help a student study and reinforce material.
3. Register Generator - Software generation tool that takes a register map in XML and generates the register code in HDL and the corresponding software header/library files (C++/Python).

Given the constraints for the course - environment, time, students' skill, we chose the Health Tracker application as the primary goal of the course is the software engineering process, and the application is easily extensible.

Selected Application - Health Tracker

The Health Tracker will be a single-user, standalone application that runs on a Windows desktop using a GUI with a database backend written in Python. The primary purpose is to track calories for the user and predict weight over time given information. The initial version will calculate BMI and generate multiple total-calorie-intake plans based on the user's BMI. Once the user selects the desired plans, the target will be set. The user records their intake of calories in the system, the system continues to accumulate the calories and lets the user know if they have reached their daily goal. Weekly/Monthly reports are presented in the form of a table or graph so that the user can track their progress.

Subsequent versions will extend on this functionality and may include: more user input to calculate body fat percentage, BMR, macro-nutrients, food entry, weight and BMI/body fat percentage predictions, and statistics for all of the above.

Software Boundaries

Platform Compatibility

The project will deliver testing for Windows 10 only. The software will be developed on a combination of macOS, Linux, and Windows platforms and is expected to be cross-platform compatible. However, given the time restriction of a single semester, testing will only be performed within Windows 10.

Functionality Mode

The app will be designed as a standalone application, primarily to limit possible complications. The app, after download, will not require a network connection to function. All data storage and retrieval processes will happen locally. The app will be designed to function for a single user only.

Software Interaction

The application will primarily interact with a local database for data storage and retrieval. There will be no integration with other external software or systems, thus avoiding any duplication of functionality, and limiting possible failure points.

User Interaction Mode

The software will utilize a graphical user interface (GUI), provisionally provided by the Tkinter library. There will be no command-line interface for the user, although other modes may exist for development and testing.

Programming Language

We expect the entire project to be developed using Python.

Database Backend

We plan on using SQLite as the database backend for user information as well as other caloric/fitness information. However, this remains subject to change as we develop the application.

Architecture

The software will strictly adhere to the traditional MVC (Model-View-Controller) architecture. This will ensure clear separation between the application's user interface, data, and control logic.