# Module Guide for SFWRENG 4G06

Team #7, Team FAAM, SweatSmart

Daniel Akselrod
Jonathan Avraham
Sophie Fillion
Sam McDonald

April 5, 2024

# 1 Revision History

| Revision Version | Date | Developer(s) | Change |
|---|---|---|---|
| 1 | April 4 2024 | Sam, Sophie, Jonathan, Daniel | Second draft of Document |

# 2 Reference Material

This section records information for easy reference.

## 2.1 Abbreviations and Acronyms

| symbol | description |
|---|---|
| AC | Anticipated Change |
| DAG | Directed Acyclic Graph |
| M | Module |
| MG | Module Guide |
| OS | Operating System |
| R | Requirement |
| SC | Scientific Computing |
| SRS | Software Requirements Specification |
| SFWRENG 4G06 | Explanation of program name |
| UC | Unlikely Change |

# Contents



# List of Tables

# List of Figures

# 3 Introduction

Breaking down a system into modules is a widely accepted strategy in software development. A module serves as a task assignment for a programmer or programming team. Our approach advocates decomposition based on the principle of information hiding. This principle aligns with designing for change, as the concealed "secrets" within each module anticipate potential future modifications. Designing for change is particularly valuable in SC, where alterations are frequent, especially during the initial development phase as the solution space is explored.

Our design follows the rules, as follows:

- System details that are likely to change independently should be the secrets of separate modules.
- Each data structure is implemented in only one module.
- Any other program that requires information stored in a module's data structures must obtain it by calling access programs belonging to that module.

After completing the first stage of the design, the Software Requirements Specification (SRS), the Module Guide (MG) is developed [*ParnasEtAl1984*]. The MG specifies the modular structure of the system and is intended to allow both designers and maintainers to easily identify the parts of the software. The potential readers of this document are as follows:

- New project members: This document can be a guide for a new project member to easily understand the overall structure and quickly find the relevant modules they are searching for.
- Maintainers: The hierarchical structure of the module guide improves the maintainers' understanding when they need to make changes to the system. It is important for a maintainer to update the relevant sections of the document after changes have been made.
- Designers: Once the module guide has been written, it can be used to check for consistency, feasibility, and flexibility. Designers can verify the system in various ways, such as consistency among modules, feasibility of the decomposition, and flexibility of the design.

The rest of the document is organized as follows. Section 4 lists the anticipated and unlikely changes of the software requirements. Section 5 summarizes the module decomposition that was constructed according to the likely changes. Section 6 specifies the connections between the software requirements and the modules. Section 7 gives a detailed description of the modules. Section 8 includes two traceability matrices. One checks the completeness of the design against the requirements provided in the SRS. The other shows the relation between anticipated changes and the modules. Section 9 describes the use relation between modules.

# 4 Anticipated and Unlikely Changes

This section lists possible changes to the system. According to the likeliness of the change, the possible changes are classified into two categories. Section 4.1 lists anticipated changes, and Section 4.2 lists unlikely changes.

## 4.1 Anticipated Changes

Anticipated changes are the source of the information that is to be hidden inside the modules. Ideally, changing one of the anticipated changes will only require changing the one module that hides the associated decision. The approach adapted here is called design for change.

**AC1:** The format of the initial input data.

**AC2:** The exercises included in the exercise list.

**AC3:** The parameters a user can edit for their profile.

**AC4:** The feedback form prompts and interaction.

**AC5:** Algorithm improvement for workout generation.

## 4.2 Unlikely Changes

The module design should be as general as possible. However, a general system is more complex. Sometimes this complexity is not necessary. Fixing some design decisions at the system architecture stage can simplify the software design. If these decision should later need to be changed, then many parts of the design will potentially need to be modified. Hence, it is not intended that these decisions will be changed.

**UC1:** Input/Output devices (Input: File and/or Keyboard, Output: File, Memory, and/or Screen).

**UC2:** The encryption scheme of data.

# 5 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 1. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

**M1:** User Profile Management Module

**M2:** Workout Generation Module

**M3:** Data Management Module

**M4:** User Interaction Module

**M5:** Reporting and Feedback Module

**M6:** ChatBot Integration Module

**M7:** Algorithm Selection Module

**M8:** Input Format Module

| **Level 1** | **Level 2** |
|---|---|
| Behaviour-Hiding | User Profile Management Module |
| | Workout Generation Module |
| | Data Management Module |
| | User Interaction Module |
| | Reporting and Feedback Module |
| | Chatbot System Module |
| Software Decision | Algorithm Selection Module |
| | Input Format Module |

Table 1: Module Hierarchy

# 6 Connection Between Requirements and Design

The design of the system is intended to satisfy the requirements developed in the SRS. In this stage, the system is decomposed into modules. The connection between requirements and modules is listed in Table 2.

# 7 Module Decomposition

The SweatSmart application is composed of several distinct modules, each responsible for a separate concern within the system. This modular approach ensures that each part of the application can be developed, tested, and maintained in isolation, promoting a scalable and robust software architecture.

The following sections detail the specific responsibilities and design of each module, providing insight into their individual functions as well as their interactions with other parts of

the system. The decomposition is designed to encapsulate core functionalities and to define clear boundaries and interfaces for interaction, facilitating a plug-and-play architecture that can evolve as new features and requirements emerge.

Each module is described with an emphasis on its "secret" – the design and knowledge it hides from other modules, its interface – the access points for other parts of the system to interact with it, and its interaction with other modules, which delineates how it cooperates within the system to deliver cohesive functionality.

The primary modules identified for the SweatSmart application are as follows:

- User Profile Module
- Workout Generation Module
- Data Management Module
- User Interaction Module
- Reporting and Feedback Module
- Algorithm Selection Module
- Input Format Module

In subsequent sections, we will delve into the particulars of each module, starting with the User Profile Module.

## 7.1 Behaviour-Hiding Module

### 7.1.1 User Profile Management Module (M1)

The User Profile Management Module is central to the personalization features of the SweatSmart application. Its primary responsibility is to manage user-specific data, which includes account creation, user authentication, and the storage and retrieval of personal fitness profiles.

**Secrets:** The module encapsulates the logic for user data validation, ensuring that data integrity is maintained and personal information is stored securely. It also handles the encryption and decryption of sensitive information to protect user privacy. Additionally, this module will include logic to provide data in the format expected by the Algorithm Selection Module for personalized workout algorithms.

**Services:**

- Account creation and management.

- User login and authentication processes.
- Personal information updates and management.
- Privacy settings and account security features.
- Services for formatting user data as per the standards defined in the Input Format Module.

**Interfaces:**

- An API endpoint for account creation and modification.
- Authentication services that integrate with third-party identity providers.

By maintaining a clear separation of concerns, the User Profile Management Module ensures that user-related features are modular, scalable, and independent, enabling effective and secure management of user-specific data within the broader SweatSmart system. This module's design will provide potential for future enhancements such as integration with social media platforms for enriched user profiles and community engagement.

#### 7.1.2 Workout Generation Module (M2)

The Workout Generation Module is the central aspect of the SweatSmart application, responsible for curating personalized workout plans tailored to the individual user's fitness goals, preferences, and progress.

**Secrets:** The module is designed to hold evidence-based algorithms that consider a range of variables such as the user's exercise history, current fitness level, performance metrics, and personal targets. It ensures that workouts are challenging and effective, promoting consistent user progress. It will refer to the Algorithm Selection Module for choosing the appropriate workout generation algorithm.

**Services:**

- Creation of personalized workout routines.
- Adaptation of workouts in response to user feedback.
- Dynamic adjustment to plans based on user feedback and workout results.
- Integration of rest and active days to optimize recovery and growth.
- Receives pre-validated and formatted data from the Input Format Module.

**Interfaces:**

- API endpoints for initiating workout generation and retrieving the latest plans.

- Interfaces to receive user data updates from the User Profile Management Module.
- Output methods to deliver the workout plan to the User Interaction Module.

Interaction with Other Modules: The Workout Generation Module interacts extensively with the User Profile Management Module to obtain necessary user data. The module also feeds data to the Reporting and Feedback Module to inform users of their progress and areas for improvement. Its algorithms can be updated or enhanced based on patterns observed from the Data Management Module, ensuring the workouts remain effective as fitness sciences evolve.

By maintaining this modular approach, the Workout Generation Module can be independently updated or replaced if new fitness planning methodologies are adopted, without the need to overhaul the entire system. This ensures that SweatSmart can remain at the cutting edge of fitness evidence, providing users with effective, science-backed workout regimens.

### 7.1.3 Data Management Module (M3)

The Data Management Module in SweatSmart is integral for handling all data storage, retrieval, and management operations. It ensures efficient and secure handling of user data, workout logs, and application metrics.

**Secrets:** The module encapsulates the database schema design, data access layers, and the data querying logic. It is responsible for ensuring data integrity, consistency, and optimizing data retrieval and storage operations. This module also interfaces with the Input Format Module to ensure all data stored and retrieved is correctly formatted.

**Services:**

- Secure storage of user profiles, workout histories, and preferences.
- Efficient retrieval of data for real-time application use.
- Management of data backups and restoration processes.
- Implementation of data privacy and protection mechanisms.
- Formats data according to the rules of the Input Format Module before storage.

**Interfaces:**

- APIs for data CRUD (Create, Read, Update, Delete) operations.
- Interfaces to interact with the User Profile Management and Workout Generation Modules for data exchange.

- Data export functions for reporting and analysis purposes.

Interaction with Other Modules: The Data Management Module serves as the backbone for the User Profile Management and Workout Generation Modules, providing them with necessary data. It also interacts with the Reporting and Feedback Module to supply data for generating user reports and analytics.

By centralizing data management, SweatSmart ensures that all modules have consistent and reliable access to the necessary data, enhancing the application's overall performance and user experience.

### 7.1.4 User Interaction Module (M4)

The User Interaction Module is a pivotal component of SweatSmart, focusing on the user interface (UI) and user experience (UX) aspects of the application. It facilitates the interaction between the user and the software's functionalities.

**Secrets:** This module contains the design elements, layout configurations, and interaction logic that define how users interact with the application. It ensures a seamless, intuitive, and engaging user experience. It also contains the logic to handle formatting feedback data from the Input Format Module to be sent to other modules.

**Services:**

- Rendering of the user interface, including screens for profile management, workout plans, and feedback.
- Handling user inputs and translating them into actions within the application.
- Displaying personalized content and workout suggestions.
- Alerting users when input data does not meet formatting requirements.

**Interfaces:**

- Front-end frameworks and libraries for UI rendering.
- API endpoints for sending user inputs to backend modules and receiving data.
- Integration with Workout Generation Module for real-time data display.

Interaction with Other Modules: The User Interaction Module is closely linked with almost all other modules. It receives data from the Data Management Module for display and sends user inputs back for processing. It also works in conjunction with the Workout Generation Module to present tailored workout plans to enhance user engagement.

By focusing on a user-centric design, this module plays a crucial role in ensuring that SweatSmart is not only functional but also enjoyable and easy to use.

### 7.1.5 Reporting and Feedback Module (M5)

The Reporting and Feedback Module is a key component of SweatSmart, focusing on the collection, analysis, and presentation of user feedback and workout data. This module is instrumental in understanding user satisfaction and system performance.

**Secrets:** The module houses the mechanisms for gathering user feedback, analyzing workout data, and generating reports. It ensures confidentiality and accuracy in feedback processing and report generation. It includes the decision-making logic provided by the Algorithm Selection Module for generating insights based on user feedback.

**Services:**

- Collection and storage of user feedback from various points within the app.
- Analysis of workout data to generate performance reports.
- Generation of insights based on user activity and feedback.
- Formats feedback data from users according to standards set by the Input Format Module.

**Interfaces:**

- Feedback collection interfaces within the app.
- Reporting APIs to deliver analyzed data and insights to users and administrators.
- Data exchange interfaces with the Data Management Module for accessing user data.

Interaction with Other Modules: The Reporting and Feedback Module interacts with the User Profile Management Module to tailor feedback requests based on user profiles. It also collaborates with the Workout Generation Module by providing insights that can be used to refine the workout algorithms. Interaction with the Data Management Module is crucial for accessing and storing the necessary data for analysis.

This module plays a vital role in driving improvements across the SweatSmart platform by translating user feedback and workout data into actionable insights.

### 7.1.6 ChatBot Integration Module (M6)

The ChatBot Integration Module serves as the intermediary between the SweatSmart application and OpenAI's Chat GPT API, enabling advanced conversational capabilities and interactive user support within the application.

**Secrets:** This module encapsulates the logic for interacting with the Chat GPT API, including constructing the HTTP requests, managing authentication, handling the conversation state, and parsing the responses. It abstracts the complexities of network communication, JSON serialization, and API key management, ensuring these implementation details are not exposed to the other parts of the system.

**Services:**

- Authenticate and communicate with the OpenAI Chat GPT API.
- Send user messages and system context to the API and receive responses.
- Serialize and deserialize chat messages to and from JSON format.
- Manage API keys securely, retrieving them from Azure Key Vault when not available in the environment variables.

**Interfaces:**

- RESTful API endpoints to receive chat messages from the client and send responses back.
- Authentication interface to secure chat endpoints.

**Interaction with Other Modules:** The ChatBot Module interacts with the User Profile Management Module to retrieve user preferences and context for personalized interactions. It also connects with the Data Management Module to store and retrieve conversation histories. By integrating with OpenAI's Chat GPT API, the ChatBot Module provides a robust solution for real-time user interaction, leveraging the capabilities of AI to deliver personalized and intelligent support within the SweatSmart application.

## 7.2 Software Decision Module

### 7.2.1 Algorithm Selection Module (M7)

The Algorithm Selection Module is a crucial software decision-making component of SweatSmart, responsible for determining the most effective algorithms for workout generation and analytics based on real-time user data.

**Secrets:** The module comprises advanced selection logic to choose the optimal algorithm for each user, considering factors like past performance, current fitness levels, and user preferences. This module also includes proprietary methods for evaluating and comparing the effectiveness of different algorithms.

**Services:**

- Dynamic selection of workout and analytics algorithms based on user data.

- Periodic evaluation of algorithm effectiveness and adaptation as necessary.
- Provision of a selection interface for other modules to request algorithm services.

**Interfaces:**

- Algorithm selection interface used by the Workout Generation and Predictive Analytics Modules.
- Interface for receiving user data and preferences from the User Profile Management Module.

Interaction with Other Modules: The Algorithm Selection Module directly impacts the functionality of the Workout Generation Module by supplying it with the appropriate algorithm for creating personalized workout plans. It also interacts with the Reporting and Feedback Module to use insights from user feedback for algorithm refinement.

By effectively selecting and managing the algorithms used across the application, the Algorithm Selection Module ensures that SweatSmart adapts to the unique and changing needs of each user, providing a highly personalized fitness experience.

### 7.2.2 Input Format Module (M8)

The Input Format Module is an integral part of SweatSmart, ensuring all user and system data is correctly formatted and validated before being processed by the application's various functional components.

**Secrets:** This module harbors the logic for data validation rules, format specifications, and sanitization procedures. It acts as the first line of defense against data inconsistencies and errors that could compromise system functionality.

**Services:**

- Validation and formatting of incoming data from user inputs and external sources.
- Sanitization of data to remove or correct any inaccuracies or inconsistencies.
- Provision of formatting rules and validation interfaces for other system modules.

**Interfaces:**

- Data validation and formatting interface utilized by the User Interaction Module.
- Sanitization interface for cleaning data received from External Integration Module.

Interaction with Other Modules: The Input Format Module serves all other modules by ensuring the data they receive is in the correct format and meets the system's data quality standards. It plays a critical role in the interaction between the User Interaction Module and the core system components, such as the Workout Generation and Predictive Analytics Modules, by providing them with clean and standardized data.

By maintaining the integrity and consistency of the data throughout SweatSmart, the Input Format Module supports robust system operations and a seamless user experience.

# 8 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

| Requirement | Modules |
|---|---|
| UA1 | M1, M3, M4, M8 |
| UA2 | M1, M3, M4, M7 |
| UA3 | M1, M3, M4, M7 |
| UA4 | M1, M4, M7 |
| UA5 | M1, M4, M7 |
| AI1 | M1, M2, M3, M4, M7 |
| AI2 | M2, M3, M7 |
| AI3 | M2, M3, M5, M7 |
| AI4 | M3, M4, M6 |
| AI5 | M3, M4, M5, M7 |
| UC1 | M1, M2, M3, M4 |
| UC2 | M4, M5 |
| UC3 | M1, M3, M4 |
| UC4 | M3 |
| WH1 | M1, M3 |
| WH2 | M3, M5 |
| DB1 | M1, M2, M3, M4, M5 |
| AF1 | M4, M7 |
| AF2 | M3, M4 |
| AF3 | M1, M2, M3, M4 |
| AF4 | M1, M3 |
| AF5 | M1, M3, M4 |

Table 2: Trace Between Requirements and Modules

| AC | Modules |
|---|---|
| AC1 | M8 |
| AC2 | M3 |
| AC3 | M1 |
| AC4 | M5 |
| AC5 | M7 |

Table 3: Trace Between Anticipated Changes and Modules

# 9 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided. *Parnas1978* said of two programs A and B that A *uses* B if correct execution of B may be necessary for A to complete the task described in its specification. That is, A *uses* B if there exist situations in which the correct functioning of A depends upon the availability of a correct implementation of B. Figure 1 illustrates the use relation between the modules. It can be seen that the graph is a directed acyclic graph (DAG). Each level of the hierarchy offers a testable and usable subset of the system, and modules in the higher level of the hierarchy are essentially simpler because they use modules from the lower levels.

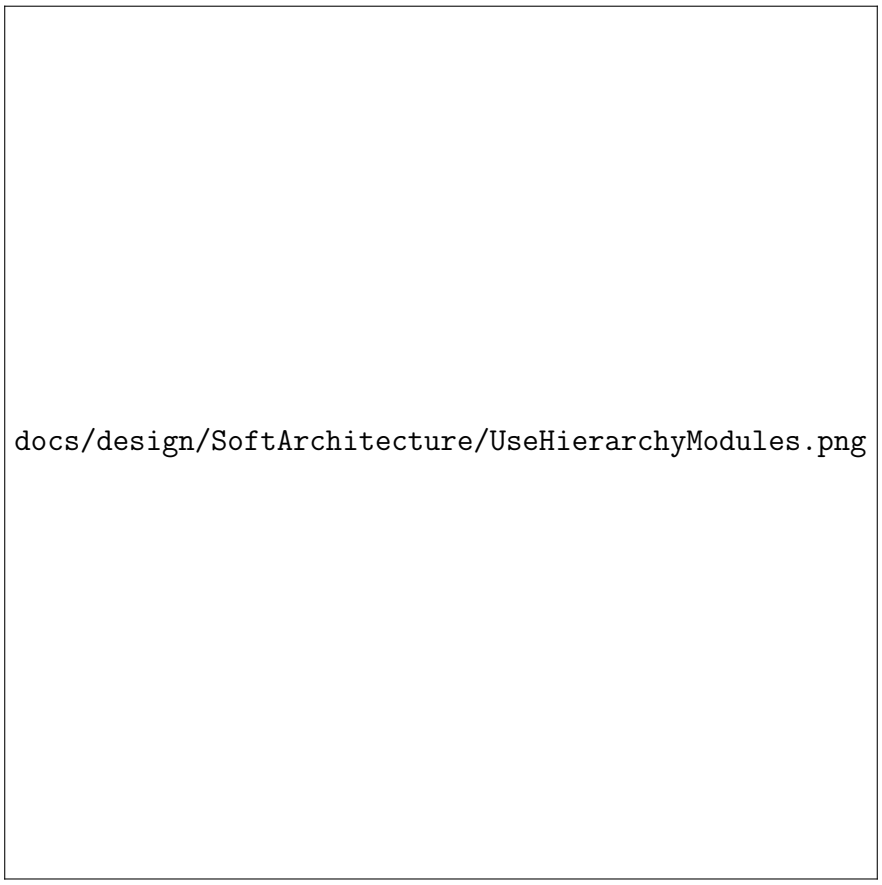


Figure 1: Use hierarchy among modules

# 10 Timeline

| Module | Developers | Deadline |
|---|---|---|
| M1 | Daniel, Jonathan | Feb 2, 2024 |
| M2 | Sam, Sophie | Jan 27, 2024 |
| M3 | Daniel, Jonathan, Sam, Sophie | Feb 2, 2024 |
| M4 | Daniel, Jonathan, Sam, Sophie | Feb 2, 2024 |
| M5 | Daniel, Jonathan, Sam, Sophie | Feb 2, 2024 |
| M6 | Daniel, Sophie | Jan 25, 2024 |
| M7 | Sam, Sophie | Jan 27, 2024 |
| M8 | Daniel, Jonathan | Jan 31, 2024 |

Table 4: Timeline for Revision 0 Demo