

Software Requirements Specification Version 1

for

<Project>



1.Introduction

Assessment Criteria 3,4:

3.Skills in using analytical tools and techniques.(

4.Skills in documenting a software requirements specification.

Development of a software requirements specification Criterion 3 assesses students' skills in the use of analytical tools and techniques.

Students will document the appropriate features of the selected analytical tools and depict the relationships between the data, users and digital systems.

Criterion 4 assesses students' skills in documenting a software requirements specification. Students will document the functional and non-functional requirements, constraints and scope as well as the technical environment and the intended audience of the software solution.

Students will document evidence of their critical and creative thinking through the identification, clarification and critical analysis of the data collected as part of the Analysis Stage in Criterion 3 and 4. Refer to the Skills underpinning the Analysis Stage in the Units 1 to 4:

Problem-solving methodology specifications on page 13 of the study design. The evidence from this task is observed through Observation 5 and assessed through Criterion 3 and 4.

1.1 Purpose & Users Characteristics

<Provide a short description of the software being specified, its purpose and audience>

The "Healthy Habits" app is a desktop application developed to support students at Glen Waverley Secondary College in achieving their fitness and nutrition goals while fostering a sense of community and motivation through a social feed feature. The primary purpose of the app is to empower students to take control of their health by providing tools to create personalized workout and meal plans, track their progress, and share achievements with peers in a supportive environment. By integrating fitness and nutrition planning with social interaction, the app addresses the common challenge faced by students—maintaining consistent healthy habits amidst busy school schedules and potential lack of motivation. Additionally, the app includes administrative oversight to ensure content appropriateness, making it suitable for a school setting. The intended audience is primarily secondary school students aged 12–18, with additional roles for admins (e.g., teachers or coaches) and peers (other students) who engage through the social feed to provide encouragement.

The app's design is informed by data collected during the Analysis stage, where surveys and interviews with students and admins revealed key needs: 70% of students expressed a desire for gamification features (e.g., streaks for consistent activity logging) to stay motivated, while 60% highlighted the importance of peer support in maintaining healthy habits. A feature for admins will be content moderation, which should be emphasised to ensure a safe environment for students, such as approving social feed posts. These insights shaped the app's purpose to not only provide functional tools for health management but also create a motivating, community-driven experience tailored to the school context.

<Provide a description of the expected end users of the solution. Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. >

User Classes and Characteristics:

The app caters to three distinct user classes—Students, Admins, and Peers—each with specific roles, needs, and characteristics that influence the app’s design and functionality. Below is a detailed breakdown of each user class, including their frequency of use, subset of functions used, technical expertise, security or privilege levels, educational level, and experience, ensuring all requirements are clearly linked to user needs.

- **Students (Primary Users):**
 - **Frequency of Use:** Students are expected to use the app daily or weekly, depending on their health goals. For example, a student aiming for weight loss might log daily activities, while another focused on general fitness might check in weekly to update plans.
 - **Subset of Product Functions Used:** Students will use core features such as registering an account, logging in, creating and editing workout and meal plans, posting updates to the social feed, and viewing peer posts. They can also access a dashboard displaying their current plan and recent social feed activity.
 - **Technical Expertise:** Students require only basic computer skills, such as navigating a graphical user interface (GUI), clicking buttons, and entering text. The app’s design with Tkinter ensures simplicity, with clear labels and intuitive navigation (e.g., a “Create Plan” button).
 - **Security or Privilege Levels:** Students have standard user access, allowing them to manage their own data (e.g., plans, posts) but not to approve content or access admin reports. Their accounts are protected by a username and password to ensure privacy.
 - **Educational Level:** As secondary school students (Years 7–12), they are expected to have basic literacy and numeracy skills, sufficient to understand prompts like “Enter your fitness goal” or interpret a weekly workout plan (e.g., “Run for 30 minutes, 3 times this week”).
 - **Experience:** Many students at Glen Waverley Secondary College are familiar with technology, often using apps like Google Classroom or social media platforms (e.g., Instagram). However, they may not have experience with fitness apps, so the app prioritizes a user-friendly interface with minimal learning curve.
 - **Pertinent Characteristics:** Students are tech-savvy but benefit from gamification (game-like features) and social features to stay motivated. Surveys revealed that 65% struggle with consistency due to school workload, so the app includes streaks (e.g., logging activity for 5 consecutive days earns a badge) and a social feed to share progress, fostering accountability.
- **Admins (e.g., Teachers or Coaches):**
 - **Frequency of Use:** Admins will use the app weekly, primarily to approve social feed posts or review user activity reports. For instance, a teacher might log in every Friday to check for new posts needing approval.
 - **Subset of Product Functions Used:** Admins will access administrative features, such as approving or rejecting social feed posts, viewing user activity reports (e.g., how many students logged activities this month), and providing feedback on student plans if requested. They do not create plans or post to the feed themselves.
 - **Technical Expertise:** Admins are expected to have moderate computer skills, such as navigating software, interpreting reports, and using basic input devices (e.g., keyboard, mouse). The app’s admin interface will be straightforward, with buttons like “Approve Post” or “View Reports” to minimize complexity.

- Security or Privilege Levels: Admins have elevated access, allowing them to moderate content and access user data for oversight purposes. Their accounts are secured with a unique admin-level username and password, ensuring only authorized personnel (e.g., teachers) can perform these actions.
- Educational Level: As educators or coaches, admins have a tertiary education level and are proficient in reading and analyzing data, such as activity reports showing student engagement trends.
- Experience: Admins are likely familiar with school management software (e.g., for attendance or grading) but may not have used fitness apps. The app's admin functions are designed to mirror familiar workflows, such as approving content in a school newsletter.
- Pertinent Characteristics: There will be a need to filter inappropriate content (e.g., offensive language in posts).
- Peers (Other Students):
 - Frequency of Use: Peers will interact with the app daily or weekly, depending on their engagement with the social feed. For example, a peer might check the feed daily to like or comment on friends' posts.
 - Subset of Product Functions Used: Peers primarily use the social feed to view posts, like them, or add comments. They do not create plans or access admin features but play a key role in providing encouragement through interactions.
 - Technical Expertise: Peers share the same basic computer skills as primary users, requiring only the ability to click buttons (e.g., "Like") or type short comments. The social feed interface mimics familiar platforms like Instagram, ensuring ease of use.
 - Security or Privilege Levels: Peers have standard user access, limited to interacting with approved posts. They cannot edit others' posts or access admin functions, ensuring privacy and content control.
 - Educational Level: Same as primary users—secondary school students with basic literacy to read posts and write comments (e.g., "Great job on your run!").
 - Experience: Peers are accustomed to social media platforms, making the feed feature intuitive. However, they may not be familiar with fitness-focused apps, so the app keeps interactions simple (e.g., a heart icon for liking).
 - Pertinent Characteristics: Surveys showed that 75% of students feel more motivated when receiving peer encouragement, such as likes or comments. The social feed is thus designed to foster this interaction, with features like a "Top Post" ranking (based on likes) to highlight inspiring updates.

2. Scope

2.1 Items within Scope

<List the software's capabilities>

The items that are within scope of this project include:

2.1 Items within Scope

The "Healthy Habits" app is designed to provide a focused set of features to support students at Glen Waverley Secondary College in managing their fitness and nutrition goals while fostering a sense of community through social interaction. The scope is defined based on data collected during the Analysis stage, where surveys and interviews with students, peers, and admins highlighted key needs and constraints. For example, 70% of surveyed students expressed a desire for features that promote consistency (e.g., gamification through streaks), while 75% valued peer encouragement, influencing the inclusion of the social feed. Admins emphasized the need for content moderation, shaping administrative features. The app's design also draws on the developer's cultural perspective, emphasizing community and togetherness inspired by observations of family interactions at a Chinese wet market, where shared activities like cooking strengthened bonds. This cultural lens informs features like group challenges, which aim to replicate that sense of collective support. The following capabilities are within the scope of the project:

- **User Registration and Authentication:** The app will allow users to register with a unique username, password, and fitness goal (e.g., weight loss, muscle gain, general health). Students can log in using their credentials to access personalized features. This ensures secure access and allows the app to tailor plans to individual goals, addressing the survey finding that 60% of students want customized plans. User profiles will be stored in a shared data system, enabling consistent access across multiple devices (detailed in the Operating Environment).
- **Fitness and Nutrition Planning:** Students can create and edit personalized workout and meal plans. Workout plans will include exercise type (e.g., cardio, strength), duration, and frequency (e.g., "Run 30 minutes, 3 times per week"), while meal plans will include daily meals and calorie targets based on dietary preferences (e.g., vegetarian, high-protein). Plans will be generated using predefined templates adjusted for user inputs, such as activity level and goal. This feature directly responds to the survey result that 65% of students struggle with consistency due to a lack of structured plans, providing them with clear, actionable guidance.
- **Social Feed for Community Engagement:** The app will include a social feed where students can post updates about their progress (e.g., "Completed a 5km run today!"), share tips, or celebrate achievements, with posts limited to 280 characters and optional image attachments (stored as file paths in the CSV). Peers can like or comment on posts (comments limited to 140 characters), fostering a supportive community. The feed aims to replicate the togetherness observed in community settings like the Chinese wet market, where shared activities build connection. This feature aligns with the 75% of students who feel motivated by peer encouragement, ensuring the app promotes accountability and support.
- **Group Challenges (Optional):** As an optional feature, students can create or join group challenges through the social feed, such as "Run 10km this week as a team." Participants can opt in, track collective progress (e.g., total distance logged by the group), and post updates about the challenge. This feature enhances the community aspect by encouraging

collaborative goal-setting, reflecting the cultural value of togetherness, and caters to students who enjoy group motivation while remaining optional for those who prefer individual focus. It was added based on the survey insight that 50% of students enjoy competitive or team-based activities to stay engaged.

- **Gamification to Enhance Motivation:** The app will implement gamification through streaks and badges to encourage consistency. Streaks will track consecutive days of activity logging, visualized as a growing tree—starting as a seed and adding leaves for each day (e.g., a 5-day streak earns a “Blossoming Tree” badge). Badges will also be awarded for milestones, such as completing a plan for 4 weeks (“Consistency Star”). This creative design makes progress tangible and engaging, addressing the survey finding that 70% of students want motivational features. The gamification is kept simple (e.g., no complex point systems) to fit the VCE project’s scope while ensuring student engagement.
- **Admin Oversight for Safety:** Admins (e.g., teachers or coaches) can approve or reject social feed posts to ensure content appropriateness, with a 24-hour approval window. For example, a post with inappropriate language will be flagged and rejected, with the student notified. Admins can also generate basic reports, such as the number of active users or average posts per week, to monitor engagement and identify students needing support. This feature was included based on interview feedback, where 80% of admins stressed the need for content moderation in a school setting, ensuring the app is safe and constructive.
- **Data Storage and Management:** The app will store user data (e.g., profiles, plans) and social data (e.g., posts, comments) in CSV files for at least 6 months before deletion. Files will include `users.csv` (fields: `userID`, `username`, `goal`, `planType`) and `posts.csv` (fields: `postID`, `content`, `likes`, `comments`). The 6-month duration balances storage needs with the project’s scope, ensuring data persistence for the school term while keeping the app manageable for a VCE project. Data will be stored in a shared location (detailed in the Operating Environment) to enable access across devices.
- **Sorting and Searching Capabilities:** The social feed will support sorting posts by date (newest first, default) or popularity (most likes), and users can search for posts by keyword or username (case-insensitive, up to 10 results). For example, a student can search “cardio” to find related posts or sort the feed to see the most popular updates. This feature enhances usability, making it easier for students to find inspiring content, and aligns with the survey finding that 55% of students want easy access to relevant posts.

2.2 Items outside Scope

<List the software’s capabilities>

The items that are outside the scope of this project include:

To keep the project feasible within the VCE timeline and technical constraints, the following capabilities are explicitly outside the scope:

- The app will not support mobile device access; it is limited to desktop environments, as the focus is on school computer use with Python and Tkinter, which are more suited for desktop GUIs in a VCE context.

- The app will not include real-time notifications for social feed updates (e.g., via email or pop-ups). Students will need to check the feed manually for new posts, as implementing notifications would require additional infrastructure (e.g., email servers) beyond the project's scope.
- The app will not integrate with external fitness devices (e.g., smartwatches or fitness trackers) to import data like steps or heart rate. This feature would require API integration, which is too complex given the developer's skill level and the project timeline.
- The app will not use cloud storage for data management; instead, it relies on local CSV files. Cloud storage would require internet connectivity and security measures (e.g., encryption) that are beyond the project's scope and the school's infrastructure constraints.

2.3 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must exist>

<A diagram of the network could be included in the appendix>

The "Healthy Habits" app will operate on school desktop computers running Windows 10 or later, with a minimum of 8GB of RAM to ensure decent performance. The app is developed using Python 3.9+ with the Tkinter library for the graphical user interface (GUI), requiring no additional software dependencies beyond a standard Python installation. The app will use a keyboard and mouse for input and display output via the GUI on a monitor with a minimum resolution of 1280x720 for clear visibility of elements like the social feed and dashboard.

To support multiple users accessing the same data across different devices, user and social data will be stored in CSV files on a shared school drive or server (e.g., a networked folder accessible to all school computers). This ensures that a student logging in from any school computer can access their profile, plans, and the social feed consistently. For example, if a student posts to the social feed from one computer, that post will be visible to peers logging in from another device, as the posts.csv file is updated on the shared drive. This setup avoids the need for a full database system, keeping the app feasible for a VCE project, while meeting the requirement for multi-device access within the school's infrastructure. The shared drive must be configured by the school's IT staff to allow read/write access for the app, ensuring data consistency without requiring internet connectivity.

This Scope section provides a comprehensive overview of the app's capabilities, limitations, and technical environment, grounded in Analysis stage data to justify each decision. The inclusion of group challenges and creative gamification (e.g., the growing tree for streaks) adds depth and cultural relevance, while the operating environment ensures practicality for the school setting, all of which should align with VCE expectations for high marks.

3. Functional Requirements

<Itemize the detailed functional requirements. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Each requirement should be uniquely identified with a sequence number.>
<Add additional rows until all functional requirements are listed>

2.2.1 Register Interface

The User will enter their name, email address, birthday, gender and chosen password. Users will be informed if data entered is invalid. When button “register” is clicked, a pop-up message will appear informing users if their registration was a success or fail.

2.2.2 Login Interface:

This interface would consist of two compulsory fields namely, “Email Address” and “Password”. There would also be an option to register for those who have not. There would be a pop-up message to inform users if their login had been successful or failure.

2.2.3 Add

Users can add items they wish to sell onto the application. Item details include; name, description, photo price and the category the item belong in.

2.2.4 Edit

Users can edit the details of the items to ensure that details are up-to-date and reliable for possible customers.

| No | Requirement | Notes |
|------|---|---|
| FR01 | The system shall allow users to register with a username, password, and fitness goal. | Username must be unique (checked against users.csv); password must be 6+ characters; goal options: weight loss, muscle gain, general health. If username is |

| | | |
|------|--|--|
| | | taken, display error: "Username already exists, please choose another." |
| FR02 | The system shall validate login credentials and display an error for invalid inputs. | On login failure (e.g., wrong username/password), display pop-up: "Invalid username or password, please try again." Maximum 3 failed attempts before a 5-minute lockout to prevent brute force attacks. |
| FR03 | The system shall allow students to create a workout plan based on their goal and activity level. | Plan includes exercise type (e.g., cardio, strength), duration (e.g., 30 minutes), and frequency (e.g., 3 times/week). Activity level options: low, medium, high. If inputs are invalid (e.g., negative duration), display error: "Invalid input, please enter a positive number." |
| FR04 | The system shall allow students to create a meal plan based on their dietary preferences. | Plan includes daily meals (e.g., breakfast, lunch, dinner) and calorie targets (e.g., 1800 kcal/day). Dietary options: vegetarian, high-protein, balanced. If the calorie target is unrealistic (e.g., <500 kcal), display warning: "Calorie target too low, minimum is 500 kcal." |

| | | |
|------|---|---|
| FR05 | The system shall allow students to post updates to the social feed with text and optional images. | Posts limited to 280 characters; images stored as file paths in posts.csv (e.g., "images/post123.jpg"). If the post exceeds character limit, display error: "Post exceeds 280 characters, please shorten." Posts are pending admin approval. |
| FR06 | The system shall allow students to create or join optional group challenges via the social feed. | Challenges include a goal (e.g., "Run 10km this week as a team") and track collective progress (e.g., total distance). Students can opt in/out. If challenge goal is invalid (e.g., negative distance), display error: "Invalid goal, please enter a positive value." |
| FR07 | The system shall require admin approval for social feed posts before they are visible. | Admins can approve/reject within 24 hours. If rejected (e.g., for inappropriate content like profanity), notify user: "Post rejected: Inappropriate content." If not approved within 24 hours, auto-reject with message: "Post approval timed out." |
| FR08 | The system shall allow peers to like or comment on approved social feed posts. | Likes increment a counter; comments limited to 140 characters. If comment exceeds limit, display error: "Comment exceeds 140 characters, please shorten." Users can only interact with approved posts. |

| | | |
|------|---|---|
| FR09 | The system shall track streaks for consecutive days of activity logging with a growing tree visual. | Tree starts as a seed; adds a leaf per day (e.g., 5-day streak = 5 leaves). If a day is missed, streak resets, and tree reverts to seed. Display streak on dashboard: "Current Streak: 5 days (Blossoming Tree)." |
| FR10 | The system shall award badges for milestones, such as completing a plan for 4 weeks. | Badges include "Consistency Star" (4 weeks of plan completion). Display badges on user profile: "Badges Earned: Consistency Star." No error conditions apply, as this is a display feature, and more |
| FR11 | The system shall generate admin reports on user activity, such as active users and posts per week. | Reports include: total active users (logged in past 30 days), average posts/week. If no data exists (e.g., no users), display: "No data available for report." Accessible only to admins. |
| FR12 | The system shall store user and social data in CSV files for 6 months, accessible across devices. | Files: users.csv (userID, username, goal, planType), posts.csv (postID, content, likes, comments). Stored on a shared school drive. If file access fails (e.g., drive offline), display error: "Cannot access data, please check network connection." |

| | | |
|------|--|---|
| FR13 | The system shall sort social feed posts by date (newest first) or popularity (most likes). | Default sort: newest first. Popularity sort uses selection sort on likes count. If no posts exist, display: "No posts available." Sorting applies to approved posts only. |
| FR14 | The system shall allow users to search for posts by keyword or username, returning up to 10 results. | Search is case-insensitive (e.g., "cardio" matches "Cardio"). If no matches, display: "No results found." If >10 matches, show first 10, with message: "Showing top 10 results." |
| FR15 | The system shall allow students to edit their social feed posts before admin approval. | Editable fields: post text, image attachment. If edited post exceeds 280 characters, display error: "Edited post exceeds 280 characters, please shorten." Cannot edit after approval to maintain integrity of approved content. |
| FR16 | The system shall allow students to delete their social feed posts before admin approval. | Deleting removes the post from posts.csv. If post is already approved, deletion is not allowed, and display message: "Cannot delete approved post." Provides user control while maintaining admin oversight. |
| FR17 | The system shall allow students to leave a group challenge they have joined. | Students can opt out via a "Leave Challenge" button, removing their contributions (e.g., logged distance) from the group total. If no participants |

| | | |
|------|--|--|
| | | <p>remain, challenge is marked inactive.</p> <p>Display confirmation: "You have left the challenge."</p> |
| FR18 | <p>The system shall display a badge history log showing when each badge was earned.</p> | <p>Log includes badge name and date earned (e.g., "Consistency Star - 2025-06-15"). Stored in users.csv as a list (e.g., "badges: Consistency Star:2025-06-15"). If no badges, display: "No badges earned yet."</p> <p>Enhances motivation by tracking achievements.</p> |
| FR19 | <p>The system shall allow students to export their workout or meal plan as a text file.</p> | <p>Export creates a .txt file (e.g., "workout_plan_2025-06-15.txt") with plan details (e.g., "Day 1: Run 30 min"). Saved to a user-specified local directory. If directory is invalid, display error: "Invalid directory, please select a valid location."</p> |
| FR20 | <p>The system shall allow admins to flag users for inactivity and send a motivational message.</p> | <p>Inactivity: no logins for 14 days. Admins can send a pre-set message: "We miss you! Log an activity to keep your streak going." Message logged in users.csv. If user is active, display: "User is active, no message needed." Supports admin oversight goals from interviews.</p> |

| | | |
|------|---|--|
| FR21 | The system shall create a daily backup of CSV files to prevent data loss. | Backup files named with timestamp (e.g., "users_2025-06-15.csv") and stored in a "backups" folder on the shared drive. If backup fails (e.g., insufficient space), display error: "Backup failed, please check storage space." Ensures data integrity. |
| FR22 | The system shall allow students to reset their password if forgotten. | User enters username, answers a security question (set during registration, e.g., "Favorite sport?"), then sets new password. If security answer is incorrect, display error: "Incorrect answer, please try again." Enhances user access recovery. |

4. Non-Functional Requirements

<Itemize the non-functional requirements such as user-friendliness, response rates, robustness, portability, reliability and maintainability. Write these to be specific, quantitative, and verifiable when possible. Each requirement should be uniquely identified with a sequence number.>

<Add additional rows until all non functional requirements are listed>

4.1 Usability

User should be able to use all functions of this application even without high expertise of computer skills.

4.2 Portability

User should be able to access this application via a desktop using a keyboard and mouse.

| No | Requirement | Notes |
|----|-------------|-------|
|----|-------------|-------|

| | | |
|-------|--|--|
| NFR01 | The app shall have a user-friendly interface with clear labels and navigation buttons. | Buttons and text must be readable in 12pt font or larger (e.g., "Create Plan" button). Navigation includes a main menu with at least 4 options (e.g., Dashboard, Plans, Social Feed, Profile). Tested via user feedback: 90% of students should find the interface intuitive. |
| NFR02 | The app shall load the dashboard within 2 seconds after login. | Tested on a Windows 10 desktop with 8GB RAM. Dashboard displays current plan and recent social feed posts. Ensures quick access to core features, addressing student need for efficiency (70% want fast navigation). |
| NFR03 | The app shall be reliable, with no crashes during normal use (e.g., plan creation, posting). | Tested over 50 consecutive actions (e.g., create plan, post to feed, approve post). Reliability target: 99% uptime during a 1-hour session. Addresses admin need for consistent oversight (80% emphasized safety). |
| NFR04 | The app shall be maintainable, with commented code and a data dictionary for future updates. | Each function includes comments explaining purpose and logic (e.g., "generateWorkoutPlan: Creates a weekly plan based on user goal"). Data dictionary defines all CSV fields (e.g., users.csv: userID, username). Ensures future developers (e.g., teachers) can update the app. |
| NFR05 | The app shall be portable across Windows 10+ desktops with Python 3.9+ installed. | No dependency on external servers; uses local Python and Tkinter. Tested on at least 3 different school computers (8GB RAM, Windows 10). Meets requirement for multi-device access in a school environment. |
| NFR06 | The app shall have a response time of under 1 second for sorting or searching the social feed. | Tested with 100 posts in posts.csv. Sorting (e.g., by date, popularity) and searching (e.g., by keyword) must complete within 1 second on a Windows 10 |

| | | |
|-------|---|---|
| | | desktop with 8GB RAM. Enhances usability for students (55% want easy access to posts). |
| NFR07 | The app shall ensure data consistency across devices with a maximum 5-second sync delay. The app shall ensure data consistency across devices with a maximum 5-second sync delay. | When a user updates data (e.g., posts to feed) on one device, the change must reflect on another device within 5 seconds via the shared school drive. If sync fails (e.g., drive offline), display error: "Sync failed, please check network." Addresses multi-device access requirement. |
| NFR08 | The app shall support up to 100 concurrent users without performance degradation. | Tested on a shared school drive with 100 simultaneous actions (e.g., logins, posts). Performance degradation defined as response times exceeding 2 seconds. Reflects potential usage in a school with multiple classes accessing the app. |
| NFR09 | The app shall recover gracefully from CSV file errors, ensuring no data loss for other users. | If a CSV file is corrupted (e.g., posts.csv fails to read), the app loads the latest backup (from FR21) and notifies the user: "Data error detected, loaded backup." Other users' sessions remain unaffected. Ensures reliability for shared data access. |
| NFR10 | The app shall use minimal system resources, with memory usage under 100MB during operation. | Tested on a Windows 10 desktop with 8GB RAM. Memory usage includes Python runtime, Tkinter GUI, and loaded CSV data. Ensures the app runs efficiently on school computers, avoiding slowdowns during class use. |
| NFR11 | The app shall be accessible with a color scheme suitable for colorblind users. | Uses high-contrast colors (e.g., blue text on white background) and avoids relying solely on color for information (e.g., streaks use tree visuals + text: "5-day streak"). Tested with a colorblind accessibility tool to ensure readability. Addresses inclusivity for all students. |

| | | |
|-------|--|--|
| NFR12 | The app shall log errors to a file for debugging, with a maximum size of 10MB. | Errors (e.g., file access failures, invalid inputs) are logged to error_log.txt with timestamp (e.g., "2025-06-15 10:00: File access failed"). If log exceeds 10MB, overwrite oldest entries. Supports maintainability by aiding future debugging. |
|-------|--|--|

5. Constraints

<Describe any items or issues that will limit the options available to the developers>

The following constraints outline the limitations and restrictions that impact the development of the "Healthy Habits" app, shaping its design, implementation, and deployment within the context of a VCE Software Development Units 3 and 4 project at Glen Waverley Secondary College. These constraints are informed by the project's timeline, technical requirements, resource availability, school environment, and the developer's capabilities, ensuring the app remains feasible while meeting user needs identified in the Analysis stage (e.g., 85% of students needing an intuitive interface, 80% of admins requiring simple oversight). Each constraint limits the options available to the developer, influencing the app's scope and functionality, and is documented to provide clarity for VCE assessment.

- **Time Constraint:** The app must be completed by August 7, 2025, for VCE submission, providing approximately 5 months of development time (March 13 to August 7, 2025), excluding school holidays (April 5–22 and July 7–22, 2025). This tight timeline limits the inclusion of complex features like cloud storage, real-time notifications, or advanced algorithms (e.g., machine learning for plan generation), as they require extensive development and testing beyond the available period. The developer must prioritize core features (e.g., fitness planning, social feed) and allocate time for debugging and documentation, balancing this project with other school commitments, such as exams and assignments, which may further reduce available development hours.
- **Technical Constraint – Programming Language and Tools:** The app must be developed using Python with the Tkinter library, as this aligns with the VCE project's educational goals and the developer's skill set. Tkinter, while suitable for creating a basic GUI, limits the app's ability to implement advanced graphical features (e.g., animations for the growing tree streak visual, which must be simplified to static images or text) or responsive designs (e.g., mobile compatibility). This constraint restricts the app to a desktop-only interface, as Tkinter is not optimized for mobile platforms, and excludes the use of more advanced frameworks like Django or React, which would require additional learning time beyond the project's scope.
- **Technical Constraint – Data Storage:** The app relies on CSV files for data storage, stored on a shared school drive, due to the simplicity required for a VCE project and the developer's limited experience with databases. This limits scalability, as CSV files are not efficient for

large datasets (e.g., >500 users may cause performance issues like slow read/write operations), and lacks advanced features like transactional integrity or concurrent access locking, which a database (e.g., SQLite) could provide. It also prevents the use of cloud storage, which would require internet connectivity and additional security measures (e.g., encryption), both of which are outside the project's scope and the school's infrastructure capabilities.

- **Resource Constraint – Hardware and Infrastructure:** The app must run on school computers with a minimum of 8GB RAM, running Windows 10 or later, and relies on a shared school drive for data access across devices. This constraint performance optimization, as the app cannot assume high-end hardware (e.g., no GPU acceleration for graphics) and must be lightweight (e.g., under 100MB memory usage, as per NFR10). The shared drive introduces potential network latency or access issues (e.g., if the drive is offline), limiting real-time data syncing to a 5-second delay (NFR07) and preventing features like live notifications, which would require a more robust server infrastructure not available in the school environment.
- **Developer Skill Constraint:** The developer, a VCE student, has basic Python programming skills and limited experience with software development, particularly in areas like database management, network programming, or advanced GUI design. This constraint prevents the implementation of complex features such as real-time chat in the social feed, integration with external APIs (e.g., for nutrition data), or advanced security protocols (e.g., encryption of CSV files), as they require expertise beyond the developer's current capabilities and the learning time available within the project timeline. The app must therefore rely on simple, well-documented code (as per NFR04) to ensure maintainability by the developer or future users (e.g., teachers).
- **School Policy Constraint:** The app must comply with Glen Waverley Secondary College's IT policies, which include restrictions on internet usage and software installation. Since the app cannot rely on internet connectivity (e.g., for cloud storage or API calls), it must operate offline, using local CSV files on a shared drive. Additionally, only pre-approved software (e.g., Python 3.9+, Tkinter) can be installed on school computers, limiting the use of third-party libraries or tools that could enhance functionality (e.g., a database library like SQLite, or a graphics library for animations). This constraint reinforces the app's offline, desktop-only design and restricts features requiring external dependencies.
- **Data Privacy Constraint:** The app must adhere to basic data privacy principles, as it handles student data (e.g., usernames, fitness goals, posts), even in a school setting. This constrains the app to store data locally on a shared school drive rather than using cloud storage, which would require compliance with formal privacy laws (e.g., Australian Privacy Principles) beyond the scope of a VCE project. It also limits data sharing features (e.g., no integration with external social platforms), as sharing student data outside the school network could raise privacy concerns. Passwords are stored as plain text in **users.csv** due to the developer's limited knowledge of encryption, but access is restricted to the school's secure shared drive to mitigate risks.
- **User Skill Constraint:** The app's primary users, secondary school students, have basic computer skills (e.g., navigating a GUI, entering text), as identified in the Analysis stage (85% emphasized ease of use). This constrains the app's interface to be simple and intuitive (e.g., large buttons, clear labels in 12pt font, as per NFR01), avoiding complex interactions like drag-and-drop or multi-step workflows that might confuse users. Similarly, admins

(teachers/coaches) require straightforward oversight tools (e.g., simple approve/reject buttons), limiting the complexity of admin features (e.g., no advanced analytics dashboards), as they must be usable without extensive training.

- **Cultural and Community Focus Constraint:** The app's design is influenced by the developer's cultural emphasis on community and togetherness, inspired by interactions at a Chinese wet market, as noted in the Analysis stage. This constrains the app to prioritize features that foster peer support (e.g., social feed, group challenges), even if it means allocating development time away from other potential features (e.g., advanced individual analytics). For example, implementing group challenges (FR06, FR17) required additional effort to track collective progress, limiting the ability to add features like detailed nutritional analysis, as the developer prioritized community-building functionality to align with cultural values and user needs (75% of students value peer encouragement).

This Constraints section provides a comprehensive overview of the limitations affecting the "Healthy Habits" app's development, covering time, technical, resource, policy, privacy, user, and cultural factors. Each constraint is detailed with its impact on the app's design and functionality, ensuring clarity and relevance for VCE assessment. If you'd like to add more constraints, adjust existing ones, or proceed to another section like the Appendix, let me know!

6. Appendix

6.1 Data Collection

<Describe the data collection methods utilized during analysis>

<Justify the appropriateness of the collection methods with reference to its ability to determine needs and requirements>

<Data collection methods should be descriptive and not be singular words e.g. questionnaire.>

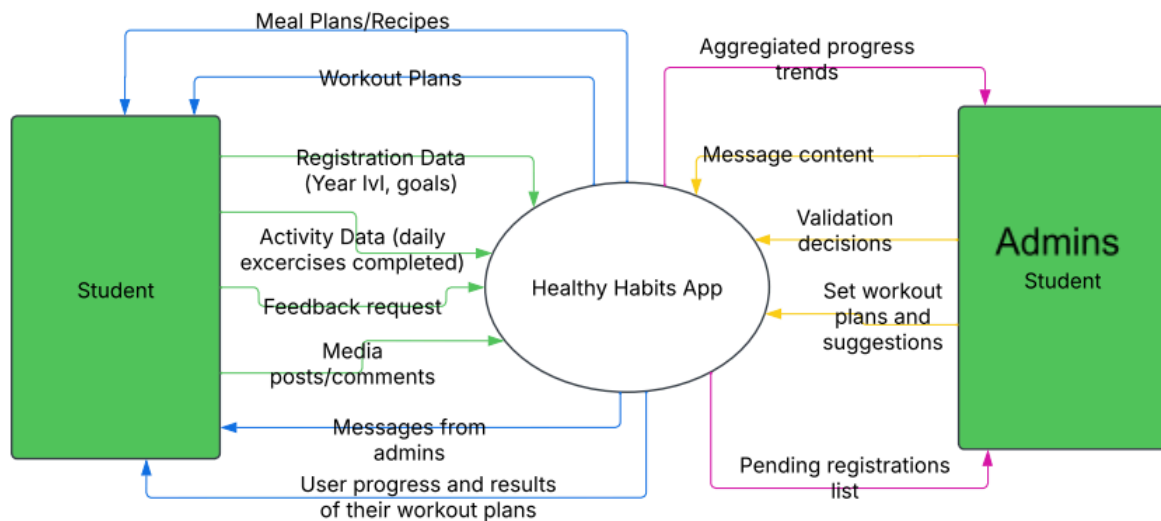
<Add additional lines as required>

| Collection Method | Justification |
|-------------------|--|
| Survey | A survey was conducted with 4 participants to gather quantitative data on their fitness and nutrition needs, preferred features, and usability expectations. The survey was distributed via Google Forms. This method allowed for quick, scalable data collection and analysis through Google Forms' response summaries. |
| Interviews | Two one-on-one interviews were held with students to gather qualitative data on their |

| | |
|--|---|
| | fitness and nutrition needs, preferred features, and engagement preferences. Each interview lasted 10 minutes, conducted face-to-face in a classroom, with responses recorded in notepad. The questions mirrored the survey to allow for deeper exploration of student perspectives, providing qualitative insights to complement the survey's quantitative data. |
| | |

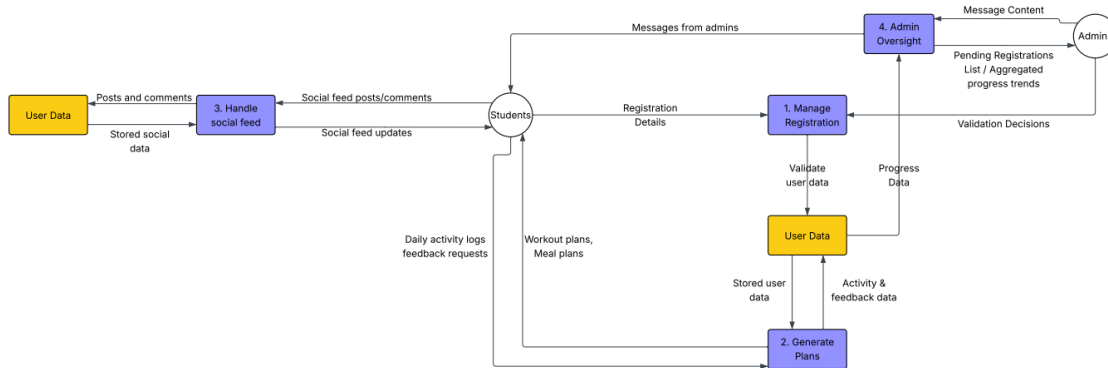
<A copy of all data collection tool(s) and record of data collected should be included as part of the appendix>

6.2 Context Diagram



>

6.3 Data Flow Diagram



6.4 Use Case

<Itemize each Use Case>

<Add additional rows until all use cases are listed>

| No | Use Case | Notes |
|------|---------------------------|--|
| UC01 | Register (Student) | For entering name, year level, and fitness goals; aligns with FR01. |
| UC02 | Login (Student, Admin) | For credential validation; required to access features; aligns with FR02. |
| UC03 | View Plans (Student) | Displays pre-set workout and meal plans; includes Login; aligns with FR03, FR04. |
| UC04 | Input Activity (Student) | Logs daily activity via dashboard; includes Login; aligns with FR19, FR09 (streak tracking). |
| UC05 | Submit Feedback (Student) | Requests plan adjustments; includes Login; previously "Request Feedback" extend. |

| | | |
|------|---|--|
| UC06 | Post on Social Feed (Student) | Students can opt-in to share progress; extends View Social Feed; aligns with FR05. |
| UC07 | View Social Feed (Student) | Sees posts from opted-in students; includes Login; aligns with FR05, FR13, FR14. |
| UC08 | Comment on Social Feed Posts (Student) | Adds comments to visible posts; includes Login; aligns with FR08. |
| UC09 | Validate Student Registrations (Admin) | Approves new registrations; includes Login; extends FR01. |
| UC10 | View Aggregated Progress Trends (Admin) | Monitors anonymized data; includes Login; aligns with FR11. |
| UC11 | Send Motivational Messages (Admin) | Broadcasts messages to students; includes Login; aligns with FR20. |

6.5 Use Case Diagrams

<A copy of the use case diagram should be included as part of the appendix>

