Software Requirements Specification (SRS)

School Club Membership Manager

Developer: Ishan

Date: April 02, 2025

Document Control

Version, Date, Changes		
1.0	02/04/2025	Initial draft of SRS
1.1	24/04/2025	Incorporated VCAA rubric feedback, added sections (e.g., Accessibility, Traceability Matrix), moved system diagrams to Appendix

Executive Summary

The **School Club Membership Manager** is a software solution designed for Glen Waverley Secondary College (GWSC) to streamline the management of school club memberships, attendance tracking, and communication. This SRS defines the system's scope, functional and non-functional requirements, constraints, and assumptions, ensuring alignment with stakeholder needs and VCAA Study Design requirements for Unit 3, Outcome 2 (U3O2). Key features include user login/registration with role-based access, club selection via a GUI, automated reminders, an admin dashboard, report generation, and an event calendar. The system will be built using Python and Tkinter, storing data in CSV files for compatibility with Compass, and must run on older school computers. The project will be completed by 03/07/2025 within budget constraints, using free tools and a single developer.

Table of Contents

1. Introduction
   1.1 Purpose
   1.2 Scope
   1.3 Stakeholders
   1.4 Definitions, Acronyms, and Abbreviations
   1.5 References
   1.6 Alignment with VCAA Study Design
2. Data Collection and Analysis
   2.1 Selected Data Collection Methods
   2.2 Data Collected from Each Method
   2.3 How Data Determines User Characteristics and Technical Environment
   2.4 Data Preparation for Analysis
   2.5 Alternative Solutions Considered
   2.6 Ethical Considerations
3. Functional Requirements
   3.1 User Authentication and Role-Based Access
   3.2 Club Selection and Management
   3.3 Attendance Tracking
   3.4 Automated Reminders
   3.5 Admin Dashboard
   3.6 Report Generation
   3.7 Event Calendar
   3.8 Data Storage and Retrieval
4. Non-Functional Requirements
   4.1 Usability
   4.2 Performance
   4.3 Security
   4.4 Compatibility
   4.5 Reliability
   4.6 Scalability
   4.7 Maintainability
   4.8 Technical Requirements

4.9 Accessibility
4.10 Portability
5. Constraints
5.1 Timeline
5.2 Budget
5.3 Hardware
5.4 Software
5.5 Personnel
5.6 Legal/Ethical Constraints
6. Assumptions
6.1 User Assumptions
6.2 Technical Assumptions
6.3 Operational Assumptions
7. System Diagrams
7.1 Use Case Diagram
7.2 Context Diagram
7.3 Data Flow Diagram (Level 0)
8. User Interface Requirements
8.1 General UI Design Principles
8.2 Specific UI Requirements
9. Testing and Validation
9.1 Testing Strategy
9.2 Validation Against Requirements
9.3 User Acceptance Testing
9.4 Test Cases
10. Risk Analysis and Mitigation
10.1 Identified Risks
10.2 Mitigation Strategies
11. Change Management
11.1 Change Request Process
11.2 Version Control
12. Project Timeline and Milestones
12.1 Gantt Chart Overview
12.2 Key Milestones
13. Glossary
14. Appendices
14.1 Interview Transcripts (Simulated)
14.2 Survey Results (Simulated)
14.3 Observation Notes
14.4 Report Analysis Summary
14.5 System Diagrams
15. Traceability Matrix
16. Conclusion
17. Submission Notes for VCAA

List of Figures

- Figure 1: Use Case Diagram (Appendix 14.5.1)
- Figure 2: Context Diagram (Appendix 14.5.2)
- Figure 3: Data Flow Diagram (Level 0) (Appendix 14.5.3)

1. Introduction

1.1 Purpose

The **School Club Membership Manager** is a software solution designed for GWSC to streamline the management of school club memberships, attendance tracking, and communication. The system addresses inefficiencies in current manual processes, such as:

- **Time-Consuming Attendance Tracking**: Club leaders spend 10–15 minutes per meeting manually recording attendance (observations).
- **Lack of Centralized Data**: Membership data is inconsistent across clubs, with some using Excel and others paper (reports).
- **Ineffective Communication**: Students often forget meetings due to a lack of reminders (interviews, observations).

The purpose of this SRS is to define the functional and non-functional requirements, constraints, and assumptions for the system, ensuring it meets the needs of all stakeholders while adhering to the VCAA Study Design requirements for Unit 3, Outcome 2 (U3O2).

1.2 Scope

In-Scope:

- User login/registration with role-based access (students, club leaders, admins, GWSC Committee).
- Club selection and attendance tracking via a graphical user interface (GUI) with checkboxes.
- Automated reminders for club meetings (in-app or email-based).
- Admin dashboard for monitoring membership and attendance data.
- Report generation (membership numbers, attendance trends, inactive members).
- Event calendar to display upcoming club events.
- Data storage and retrieval using CSV files for compatibility with Compass.

Out-of-Scope:

- Direct integration with Compass (data will be exported to CSV for manual import).
- SMS notifications (reminders will be in-app or email-based).
- Financial transactions (e.g., club fees).
- Mobile app development (system will be desktop-based).
- Real-time chat functionality between users.

Future Enhancements:

- The system may support SMS notifications and direct Compass integration in a future version, pending budget approval.
- Multilingual support for non-English speakers may be added to enhance accessibility.

1.3 Stakeholders

- **GWSC Extracurricular Committee (Client)**: Oversees all club activities, requires reporting and oversight tools (e.g., admin dashboard, reports).
- **Club Leaders**: Manage individual clubs, need efficient tools for attendance tracking and communication (e.g., checkboxes, reminders).
- **Students**: End-users who join clubs, need an easy-to-use interface and reminders to attend meetings.
- **Administrator**: Manages school systems (e.g., Compass), provides insights on integration, security, and performance requirements.
- **Developer (Ishan)**: Responsible for designing, developing, and testing the system within the given constraints.

1.4 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification.
- **GUI**: Graphical User Interface.
- **CSV**: Comma-Separated Values (file format for data storage).
- **Compass**: School management system used by GWSC.
- **GWSC**: Glen Waverley Secondary College.
- **FR**: Functional Requirement.
- **NFR**: Non-Functional Requirement.
- **U3O2**: Unit 3, Outcome 2 (VCAA Study Design).
- **VCAA**: Victorian Curriculum and Assessment Authority.
- **Admin**: Refers to both school administrators and GWSC Committee members unless specified.

1.5 References

- VCAA Study Design for Applied Computing 2025–2028 (web ID: 17).
- Gantt chart for project timeline
- System diagrams (Use Case, Context, Data Flow) submitted as PNG files.

1.6 Alignment with VCAA Study Design

This SRS aligns with the VCAA Study Design for U3O2 by defining the problem (inefficient club management, Section 1.1), identifying solution requirements through data collection (Section 2), and documenting functional/non-functional requirements (Sections 3–4). Analytical tools (system diagrams) are included in Appendix 14.5, meeting the skills outlined on page 18 of the Study Design, such as "defining the problem" and "identifying solution requirements."

---

2. Data Collection and Analysis

2.1 Selected Data Collection Methods

The following methods were used to gather data, exceeding the VCAA requirement of three methods (web ID: 17):

- **Interviews**:
  - **Purpose**: Gathered qualitative data on user needs, pain points, and technical requirements.
  - **Rationale**: Direct conversations with stakeholders (GWSC Committee, club leaders, students, admin) provided detailed insights into specific needs (e.g., admin dashboard, reminders) and constraints (e.g., CSV compatibility, old hardware). Follow-up questions clarified requirements.
  - **Effectiveness**: High—revealed nuanced needs (e.g., role-based access) that surveys might miss.
- **Surveys**:
  - **Purpose**: Collected quantitative data on user preferences and pain points.
  - **Rationale**: Surveys allowed for a broader sample (students, club leaders) to quantify needs (e.g., 85% of students want reminders) and validate interview findings. Closed-ended questions provided measurable data, while open-ended questions captured additional suggestions.
  - **Effectiveness**: Medium-high—confirmed trends (e.g., preference for GUI) but lacked the depth of interviews.
- **Observations**:
  - **Purpose**: Observed real-world processes to identify inefficiencies and user behaviors.
  - **Rationale**: Watching club meetings (Chess Club, Drama Club) provided firsthand insight into manual processes (e.g., 10–15 minutes on attendance) and user struggles (e.g., Compass usability issues). This method validated interview and survey data with objective observations.
  - **Effectiveness**: High—grounded the analysis in real user interactions, highlighting practical issues.
- **Reports**:

  - **Purpose**: Analyzed existing data to identify gaps and inefficiencies.
  - **Rationale**: Reviewing club reports and Compass data revealed systemic issues (e.g., inconsistent data formats, 20% inactivity rate) that informed the solution's scope (e.g., need for centralized data storage).
  - **Effectiveness**: Medium—provided historical data but lacked user perspectives, making it less dynamic than interviews or observations.

2.2 Data Collected from Each Method

- **Interviews**:
  - **GWSC Committee**: Need an admin dashboard, reports on membership and attendance, system load time < 2 seconds.
  - **Club Leaders**: Manual attendance tracking is time-consuming (10–15 minutes per meeting); need checkboxes and reminders for meetings.
  - **Students**: Often forget meetings due to lack of reminders; need an easy club selection interface.
  - **Admin**: Requires CSV compatibility for Compass integration, role-based access for security, and the system must handle 300 users simultaneously.
- **Surveys (Simulated Results)**:
  - 85% of students want reminders for club meetings.
  - 70% of club leaders rate attendance tracking as 4/5 time-consuming.
  - 90% of users prefer a graphical interface over text-based.
  - 60% of students check club updates weekly, indicating a need for better communication.
  - 40% of students reported difficulty navigating Compass, supporting the need for a simpler interface.
- **Observations**:
  - **Chess Club Meeting**: Leaders spend 10–15 minutes manually recording attendance on paper; students arrive late or miss meetings due to lack of reminders.
  - **Drama Club Meeting**: No centralized system for club sign-ups; students ask leaders directly, leading to confusion.
  - **General Observation**: Some students struggle to use Compass on school computers (e.g., slow load times, unintuitive navigation), indicating usability issues.
- **Reports**:
  - Membership data is inconsistent across clubs (e.g., some use Excel, others paper).
  - Attendance records show 20% of members are inactive (missed 3+ meetings), highlighting the need for better engagement tools.
  - Compass lacks club-specific features, forcing manual data entry, which is error-prone.

2.3 How Data Determines User Characteristics and Technical Environment

- **User Characteristics**:
  - **Tech-Savviness**:
    - Club leaders rated themselves 4/5, students 3/5 (interviews).
    - Observations showed some students struggling with Compass (e.g., 40% reported difficulty in surveys), indicating a need for a simple, intuitive GUI.
  - **Usage Patterns**:
    - Students check updates weekly (60% from surveys), and many forget meetings (interviews, observations), justifying the need for reminders and a calendar feature.
    - Club leaders spend 10–15 minutes on attendance (observations), indicating a need for efficiency (e.g., checkboxes).
  - **Roles and Needs**:
    - Students: Need to join clubs easily, receive reminders, and mark attendance.
    - Club Leaders: Need quick attendance tracking (checkboxes) and communication tools (reminders).
    - Admins/GWSC Committee: Need oversight via a dashboard and detailed reports (e.g., membership numbers, inactive members).
  - **Diverse Needs**:
    - Some students may have visual impairments, requiring high-contrast mode and screen reader support.
    - Non-English speakers may need multilingual support (future enhancement).
  - **Training**:
    - The system shall include a user manual and a 30-minute training session for club leaders and admins to ensure effective use, given the varying tech-savviness levels.
- **Technical Environment**:
  - **Hardware**: School computers are "a bit old" (admin interview), with at least 2 GHz CPU, 4 GB RAM, 500 MB free disk space, running Windows 10, so the system must be lightweight (Python, Tkinter, CSV storage).
  - **Software**: CSV compatibility required for Compass integration (admin interview, reports analysis).
  - **Performance**: Must handle 300 users simultaneously with a load time < 2 seconds (admin interview).
  - **Security**: Role-based access and password protection needed to protect student data (admin, GWSC Committee interviews).
  - **Network**: School Wi-Fi is available but can be slow during peak times (admin interview), so the system should minimize network dependency (local CSV storage).

2.4 Data Preparation for Analysis

The collected data was labeled and categorized to prepare it for analysis, ensuring it directly informs the system requirements.

Category	Data Source	Data Point	Label
User Needs	Interviews (Students)	85% want reminders; need easy club selection	Functional Requirement
User Needs	Surveys	90% prefer graphical interface	Usability Requirement
User Needs	Observations	Students forget meetings; struggle with Compass	Functional/Usability Need
Club Leader Needs	Interviews (Leaders)	Manual attendance is time-consuming; need checkboxes	Functional Requirement
Club Leader Needs	Surveys	70% rate attendance tracking as 4/5 time-consuming	Pain Point
Club Leader Needs	Observations	Leaders spend 10–15 minutes on attendance	Efficiency Need
Admin Needs	Interviews (GWSC)	Need admin dashboard, reports on membership/attendance	Functional Requirement

Admin Needs	Interviews (Admin)	CSV compatibility, role-based access	Technical/Security Requirement
Technical Environment	Interviews (Admin)	Old school computers; handle 300 users; load time < 2 seconds	Performance Constraint
Technical Environment	Reports	Compass lacks club features; data in inconsistent formats	Technical Gap
Pain Points	Observations	No centralized sign-up; manual processes lead to confusion	System Need
Pain Points	Reports	20% of members inactive; inconsistent data formats	Efficiency Gap

2.5 Alternative Solutions Considered

- **Database vs. CSV**: A database (e.g., SQLite) was considered instead of CSV files for data storage but rejected due to hardware constraints (older school computers) and the need for a lightweight system. CSV files were chosen for simplicity and compatibility with Compass.
- **SMS vs. Email Reminders**: SMS reminders were considered to address the issue of students forgetting meetings but excluded due to budget constraints (no funds for SMS APIs). In-app and email-based reminders were chosen as cost-effective alternatives.

2.6 Ethical Considerations

- **Student Data Privacy**: Interviews confirmed the need for role-based access (admin, GWSC Committee) to prevent unauthorized access to student data, aligning with the Australian Privacy Principles (APPs). Consent must be obtained for collecting and storing student data.
- **Transparency**: The system will inform users about how their data is used (e.g., for attendance tracking, reminders) through a privacy notice during registration.

3. Functional Requirements

Functional requirements define the specific features and functionalities the system must provide. Each requirement includes a priority level (High/Medium/Low) to indicate importance.

3.1 User Authentication and Role-Based Access

- **FR1.1**: The system shall allow users to register with a username and password.
    - **Priority**: High
    - **Error Handling**: The system shall display an error message if the username is already taken.
- **FR1.2**: The system shall allow users to log in using their credentials.
    - **Priority**: High
    - **Error Handling**: The system shall display an error message if the credentials are invalid.
- **FR1.3**: The system shall assign roles to users (Student, Club Leader, Admin/GWSC Committee) during registration.
    - **Priority**: High
    - **Error Handling**: The system shall prevent registration if a role is not selected.
- **FR1.4**: The system shall restrict access to features based on user roles:
    - Students: Can select clubs, mark attendance, view calendar, receive reminders.
    - Club Leaders: Can mark attendance, manage club details, receive reminders.
    - Admins/GWSC Committee: Can view dashboard, generate reports, manage all data.
    - **Priority**: High
    - **Error Handling**: The system shall display an error if a user attempts to access an unauthorized feature (e.g., "Access Denied").
- **FR1.5**: The system shall validate user inputs during registration to ensure usernames are unique and passwords meet minimum security criteria (e.g., 8 characters, including a number).
    - **Priority**: High
    - **Error Handling**: The system shall display an error if validation fails (e.g., "Password must be at least 8 characters").

3.2 Club Selection and Management

- **FR2.1**: The system shall display a list of available clubs for students to select from.
    - **Priority**: High
    - **Error Handling**: The system shall display a message if no clubs are available.
- **FR2.2**: The system shall allow students to join or leave clubs via a GUI with checkboxes.
    - **Priority**: High
    - **Error Handling**: The system shall prevent students from joining a club if the maximum capacity is reached (e.g., 30 members).

- **FR2.3**: The system shall allow club leaders to update club details (e.g., meeting times, description).
    - **Priority**: Medium
    - **Error Handling**: The system shall display an error if the updated meeting time conflicts with another event.
- **FR2.4**: The system shall store club membership data in a CSV file.
    - **Priority**: High
    - **Error Handling**: The system shall log an error if the CSV file cannot be written to.

3.3 Attendance Tracking

- **FR3.1**: The system shall allow students and club leaders to mark attendance for club meetings using checkboxes in a GUI.
    - **Priority**: High
    - **Error Handling**: The system shall display an error if attendance is marked for a past meeting date.
- **FR3.2**: The system shall record attendance data (e.g., date, student ID, club ID) in a CSV file.
    - **Priority**: High
    - **Error Handling**: The system shall log an error if the CSV file is inaccessible.
- **FR3.3**: The system shall allow club leaders to view attendance history for their club.
    - **Priority**: Medium
    - **Error Handling**: The system shall display a message if no attendance records are available.

3.4 Automated Reminders

- **FR4.1**: The system shall send automated reminders to students and club leaders for upcoming club meetings.
    - **Priority**: High (85% of students want reminders)
    - **Error Handling**: The system shall log an error if a reminder cannot be sent (e.g., email server failure).
- **FR4.2**: Reminders shall be displayed in-app (e.g., pop-up notification) and can be emailed if configured.
    - **Priority**: High
    - **Error Handling**: The system shall display a message if email configuration is not set up.
- **FR4.3**: The system shall allow users to set reminder preferences (e.g., 1 day before, 1 hour before).
    - **Priority**: Medium
    - **Error Handling**: The system shall default to 1 day before if preferences are not set.

3.5 Admin Dashboard

- **FR5.1**: The system shall provide an admin dashboard for Admins and GWSC Committee members.
  - **Priority**: High
  - **Error Handling**: The system shall display an error if dashboard data cannot be retrieved.
- **FR5.2**: The dashboard shall display:
  - Total number of clubs.
  - Total membership numbers per club.
  - Attendance trends (e.g., average attendance per club).
  - Inactive members (missed 3+ meetings).
  - **Priority**: High
  - **Error Handling**: The system shall display a message if no data is available for a specific metric.

3.6 Report Generation

- **FR6.1**: The system shall allow Admins and GWSC Committee members to generate reports.
  - **Priority**: High
  - **Error Handling**: The system shall display an error if report generation fails.
- **FR6.2**: Reports shall include:
  - Membership numbers per club.
  - Attendance trends over a specified period.
  - List of inactive members (missed 3+ meetings).
  - **Priority**: High
  - **Error Handling**: The system shall display a message if no data is available for the selected period.
- **FR6.3**: Reports shall be exportable to CSV for integration with Compass.
  - **Priority**: High
  - **Error Handling**: The system shall log an error if the CSV export fails.

3.7 Event Calendar

- **FR7.1**: The system shall display an event calendar showing upcoming club meetings.
  - **Priority**: Medium
  - **Error Handling**: The system shall display a message if no events are scheduled.
- **FR7.2**: The calendar shall allow students to view events for their joined clubs.
  - **Priority**: Medium
  - **Error Handling**: The system shall display a message if the student has not joined any clubs.
- **FR7.3**: The system shall allow club leaders to add or update meeting events.
  - **Priority**: Medium

    - **Error Handling**: The system shall display an error if the event date is in the past.

3.8 Data Storage and Retrieval

- **FR8.1**: The system shall store all data (user credentials, club details, attendance, events) in CSV files.
    - **Priority**: High
    - **Error Handling**: The system shall log an error if a CSV file cannot be created or accessed.
- **FR8.2**: The system shall retrieve data from CSV files for display and processing.
    - **Priority**: High
    - **Error Handling**: The system shall display a message if data retrieval fails.
- **FR8.3**: The system shall ensure data consistency (e.g., no duplicate entries) during storage.
    - **Priority**: High
    - **Error Handling**: The system shall log an error if duplicate entries are detected.

---

4. Non-Functional Requirements

Non-functional requirements define the quality attributes and constraints of the system.

4.1 Usability

- **NFR1.1**: The system shall have an intuitive GUI with clear menus, buttons, and labels (90% of users prefer graphical interface, per surveys).
- **NFR1.2**: The system shall require no more than 3 clicks to perform any action (e.g., mark attendance, view calendar).
- **NFR1.3**: The system shall include tooltips or help text for complex features (e.g., report generation).
- **NFR1.4**: The system shall be usable by users with basic tech skills (students: 3/5, club leaders: 4/5, per interviews).

4.2 Performance

- **NFR2.1**: The system shall have a load time of under 2 seconds for all screens (admin interview).
- **NFR2.2**: The system shall handle 300 users simultaneously without performance degradation (admin interview).
- **NFR2.3**: The system shall process attendance updates for 300 users within 5 seconds.

4.3 Security

- **NFR3.1**: The system shall implement password protection for user accounts (admin interview).
- **NFR3.2**: The system shall enforce role-based access to restrict unauthorized access (admin, GWSC Committee interviews).
- **NFR3.3**: The system shall store passwords in an encrypted format (e.g., hashed using a secure algorithm like SHA-256).
- **NFR3.4**: The system shall log all access attempts for auditing purposes.

4.4 Compatibility

- **NFR4.1**: The system shall export data to CSV files for integration with Compass (admin interview).
- **NFR4.2**: The system shall run on Windows 10, the operating system used by school computers (admin interview).
- **NFR4.3**: The system shall be compatible with Python 3.9+ and Tkinter for GUI development.

4.5 Reliability

- **NFR5.1**: The system shall have an uptime of 99.9% during school hours.
- **NFR5.2**: The system shall recover from crashes within 10 seconds by saving data to CSV files periodically.
- **NFR5.3**: The system shall handle invalid inputs (e.g., incorrect login credentials) gracefully with error messages.
- **NFR5.4**: The system shall automatically back up CSV files every 30 minutes during active use.
- **NFR5.5**: The system shall provide a manual recovery option to restore data from the latest backup.

4.6 Scalability

- **NFR6.1**: The system shall support up to 500 users in future iterations without requiring major redesign.
- **NFR6.2**: The system shall handle an increase in club numbers (up to 50 clubs) without performance issues.

4.7 Maintainability

- **NFR7.1**: The system shall be modular, with separate modules for authentication, attendance, and reporting.
- **NFR7.2**: The system shall include inline code comments for all major functions.
- **NFR7.3**: The system shall be documented with a user manual and developer guide.

4.8 Technical Requirements

- **NFR8.1**: The system shall be built using Python and Tkinter for lightweight performance on older school computers (admin interview).
- **NFR8.2**: The system shall use CSV files for data storage to minimize resource usage.
- **NFR8.3**: The system shall require no more than 100 MB of disk space for installation and data storage.

4.9 Accessibility

- **NFR9.1**: The system shall support high-contrast mode for users with visual impairments.
- **NFR9.2**: The system shall be compatible with screen readers for accessibility.

4.10 Portability

- **NFR10.1**: The system shall be designed to allow future deployment on other operating systems (e.g., macOS) with minimal changes.

---

5. Constraints

Constraints define the limitations within which the system must be developed.

5.1 Timeline

- **C1.1**: The project must be completed by 03/07/2025 for deployment in the next school term (Gantt chart milestone).
- **C1.2**: Development must align with the 16-week timeline outlined in the Gantt chart.

5.2 Budget

- **C2.1**: The project must use free tools (Python, Tkinter, Graphviz) due to budget constraints.
- **C2.2**: No external paid services (e.g., cloud hosting, SMS APIs) can be used.

5.3 Hardware

- **C3.1**: The system must run on older school computers with limited resources (e.g., 2 GHz CPU, 4 GB RAM, 500 MB free disk space, Windows 10) (admin interview).
- **C3.2**: The system must not require additional hardware purchases.

5.4 Software

- **C4.1**: The system must be compatible with existing school software (e.g., Compass via CSV export).

- **C4.2**: The system must use Python 3.9+ and Tkinter, which are already installed on school computers.

5.5 Personnel

- **C5.1**: The project is developed by a single developer (Ishan), limiting the scope to manageable features and increasing the risk of delays due to unforeseen circumstances (e.g., illness).
- **C5.2**: No external developers or consultants can be hired due to budget constraints.

5.6 Legal/Ethical Constraints

- **C6.1**: The system must comply with the Australian Privacy Principles (APPs) to protect student data, including obtaining consent for data collection.
- **C6.2**: The system must ensure that student data is not shared with unauthorized parties.

---

6. Assumptions

Assumptions outline the conditions assumed to be true for the project.

6.1 User Assumptions

- **A1.1**: Users have basic tech skills (students: 3/5, club leaders: 4/5, per interviews).
- **A1.2**: Users are familiar with basic GUI interactions (e.g., clicking buttons, selecting checkboxes).
- **A1.3**: Users will provide accurate data during registration (e.g., correct role selection).

6.2 Technical Assumptions

- **A2.1**: School computers have Python 3.9+ and Tkinter installed (admin confirmation).
- **A2.2**: School Wi-Fi is available but not required for core functionality (local CSV storage).
- **A2.3**: CSV files can be manually imported into Compass by the admin without additional tools.

6.3 Operational Assumptions

- **A3.1**: The system will be used primarily during school hours, with peak usage during lunch (300 users, admin interview).
- **A3.2**: Club meetings occur weekly, with an average of 5 meetings per club per term (reports).
- **A3.3**: The school provides access to club reports and Compass data for analysis (admin confirmation).

7. System Diagrams

System diagrams provide visual representations of the system's structure and interactions. Detailed descriptions and diagrams are included in Appendix 14.5.

7.1 Use Case Diagram

The Use Case Diagram illustrates the interactions between actors and the system. See Appendix 14.5.1 for the diagram and detailed description.

7.2 Context Diagram

The Context Diagram provides a high-level view of the system and its interactions with external entities. See Appendix 14.5.2 for the diagram and detailed description.

7.3 Data Flow Diagram (Level 0)

The Data Flow Diagram (Level 0) breaks down the system into major processes and data flows. See Appendix 14.5.3 for the diagram and detailed description.

8. User Interface Requirements

8.1 General UI Design Principles

- **UI1.1**: The system shall use a consistent color scheme (e.g., blue and white) to align with GWSC branding.
- **UI1.2**: The system shall use a font size of at least 12pt for readability (considering older school monitors).
- **UI1.3**: The system shall provide visual feedback for actions (e.g., button press changes color).
- **UI1.4**: The system shall follow the principle of least surprise (e.g., standard icons for save, delete).

8.2 Specific UI Requirements

- **UI2.1 (Login Screen)**:
  - Fields for username and password.
  - "Login" and "Register" buttons.
  - Error message display for invalid credentials.

- **UI2.2 (Student Dashboard)**:
    - List of available clubs with checkboxes to join/leave.
    - Calendar widget showing upcoming events.
    - Notification area for reminders.
- **UI2.3 (Club Leader Dashboard)**:
    - List of club members with checkboxes to mark attendance.
    - Button to add/update meeting events.
    - Notification area for reminders.
- **UI2.4 (Admin Dashboard)**:
    - Summary statistics (total clubs, membership numbers, attendance trends).
    - Button to generate reports.
    - Dropdown to select report type (membership, attendance, inactive members).
- **UI2.5 (Error Handling)**:
    - Display error messages in red (e.g., "Invalid username or password").
    - Provide a "Help" button linking to a user manual.

---

9. Testing and Validation

9.1 Testing Strategy

- **T1.1 (Unit Testing)**: Test each module (e.g., authentication, attendance tracking) independently using Python's `unittest` framework.
- **T1.2 (Integration Testing)**: Test interactions between modules (e.g., attendance data → reminders).
- **T1.3 (System Testing)**: Test the entire system against functional and non-functional requirements (e.g., load time < 2 seconds).
- **T1.4 (Performance Testing)**: Simulate 300 users using a load testing tool (e.g., `locust`) to verify performance requirements.

9.2 Validation Against Requirements

- **V1.1**: Verify that all functional requirements (FR1–FR8) are implemented and working as specified.
- **V1.2**: Verify that all non-functional requirements (NFR1–NFR10) are met:
    - **Usability (NFR1)**: Conduct user testing with a sample of students (3/5 tech-savviness) and club leaders (4/5 tech-savviness) to ensure the GUI is intuitive and actions take no more than 3 clicks.
    - **Performance (NFR2)**: Measure load time (< 2 seconds) and simulate 300 concurrent users to ensure no degradation.
    - **Security (NFR3)**: Test role-based access by attempting unauthorized actions (e.g., a student accessing the admin dashboard) and verify password encryption.

- **Compatibility (NFR4)**: Export data to CSV and manually import into Compass to confirm compatibility.
    - **Reliability (NFR5)**: Simulate crashes and verify recovery within 10 seconds; test invalid inputs to ensure error messages are displayed.
    - **Scalability (NFR6)**: Test with 500 users and 50 clubs to ensure the system scales without issues.
    - **Maintainability (NFR7)**: Review code modularity and documentation to ensure future updates are feasible.
    - **Technical Requirements (NFR8)**: Confirm the system runs on older school computers (Windows 10, 4 GB RAM) using Python 3.9+ and Tkinter.
    - **Accessibility (NFR9)**: Verify high-contrast mode and screen reader compatibility.
    - **Portability (NFR10)**: Ensure the system design allows future deployment on other OS with minimal changes.
- **V1.3**: Validate that the system addresses stakeholder needs identified through data collection:
    - Students: Easy club selection, reminders, and calendar (FR2, FR4, FR7).
    - Club Leaders: Quick attendance tracking and communication tools (FR3, FR4).
    - Admins/GWSC Committee: Dashboard and reports (FR5, FR6).
- **V1.4**: Ensure all constraints (C1–C6) are adhered to during development and testing (e.g., timeline, budget, hardware compatibility).

9.3 User Acceptance Testing (UAT)

- **UAT1.1**: Conduct UAT with representatives from each stakeholder group:
    - Students: Test club selection, attendance marking, reminder notifications, and calendar view.
    - Club Leaders: Test attendance tracking, club management, and reminder receipt.
    - Admins/GWSC Committee: Test dashboard functionality, report generation, and data export to CSV.
- **UAT1.2**: Use a feedback form to rate usability (1–5 scale) and functionality:
    - Target: Average score of 4/5 or higher for each stakeholder group.
    - Example Questions: “Was the club selection process easy to use?” “Did the dashboard provide useful insights?”
- **UAT1.3**: Address any issues identified during UAT by iterating on the system design (e.g., adjust UI elements, fix bugs).
- **UAT1.4**: Obtain sign-off from the GWSC Extracurricular Committee to confirm the system meets their requirements.

9.4 Test Cases

- **Test Case 1 (FR1.1)**: Register a new user with a unique username and password.
    - **Expected Result**: User is registered successfully, and role is assigned.
- **Test Case 2 (FR4.1)**: Schedule a club meeting and verify that a reminder is sent to students and club leaders.

    - **Expected Result**: Reminder is displayed in-app 1 day before the meeting.
- **Test Case 3 (FR6.3)**: Generate a report and export it to CSV.
    - **Expected Result**: Report is exported as a CSV file compatible with Compass.
- **Test Case 4 (NFR1.2)**: Perform an action (e.g., mark attendance) and count the number of clicks.
    - **Expected Result**: Action is completed within 3 clicks.
- **Test Case 5 (NFR3.2)**: Attempt to access the admin dashboard as a student.
    - **Expected Result**: Access is denied, and an error message is displayed.

---

10. Risk Analysis and Mitigation

10.1 Identified Risks

Risks are quantified using a risk matrix (Likelihood × Impact = Risk Score, out of 25).

R	Description	Likelihood (1–5)	Impact (1–5)	Risk Score	Priority
R	Older school computers may not handle 300 concurrent users efficiently	3 (Medium)	5 (High)	15	High
R	Students with low tech-savviness (3/5) may find the GUI confusing	3 (Medium)	3 (Medium)	9	Medium
R	CSV files may become corrupted or inconsistent during frequent updates	2 (Low)	5 (High)	10	Medium

R	The project may not be completed by 03/07/2025 due to unforeseen delays	3 (Medium)	5 (High)	15	High
R	Unauthorized access to student data if role-based access fails	2 (Low)	5 (High)	10	Medium

10.2 Mitigation Strategies

- **R1 Mitigation**: Optimize the system for low resource usage (e.g., minimize memory footprint, use CSV instead of a database). Conduct performance testing early to identify bottlenecks.
- **R2 Mitigation**: Include tooltips, help text, and a user manual (UI1.3, NFR7.3). Conduct usability testing during development to gather feedback and iterate on the GUI design.
- **R3 Mitigation**: Implement data validation and backup mechanisms (e.g., periodic CSV backups, check for duplicates before writing). Use file locking to prevent concurrent write issues.
- **R4 Mitigation**: Follow the Gantt chart timeline strictly, with weekly progress checks. Allocate buffer time for unexpected delays (e.g., 1-week buffer before 03/07/2025).
- **R5 Mitigation**: Test role-based access thoroughly (V1.2). Encrypt passwords and log all access attempts (NFR3.3, NFR3.4). Conduct a security audit before deployment.

11. Change Management

11.1 Change Request Process

- **CR1.1**: Stakeholders (GWSC Committee, club leaders, students, admin) can submit change requests via a form:
  - Fields: Requester name, role, description of change, reason, urgency (low/medium/high).
- **CR1.2**: The developer (Ishan) will review change requests weekly and assess their impact on scope, timeline, and budget.
- **CR1.3**: Approved changes will be prioritized based on urgency and incorporated into the development schedule.
- **CR1.4**: Rejected changes will be communicated back to the requester with a rationale (e.g., out of scope, budget constraints).

11.2 Version Control

- **VC1.1**: Use Git for version control to track changes to the codebase.
- **VC1.2**: Create branches for major features (e.g., `feature/auth`, `feature/attendance`) to isolate development.
- **VC1.3**: Tag releases with version numbers (e.g., v1.0 for initial deployment, v1.1 for post-UAT updates).
- **VC1.4**: Maintain a Changelog documenting all updates, bug fixes, and new features.

---

12. Project Timeline and Milestones

12.1 Gantt Chart Overview

The project timeline spans 16 weeks, from 01/04/2025 to 03/07/2025, as outlined in the previously submitted Gantt chart.

12.2 Key Milestones

- **Milestone 1 (Week 2, 07/04/2025)**: Complete data collection and analysis (interviews, surveys, observations, reports).
- **Milestone 2 (Week 4, 21/04/2025)**: Finalize SRS document and system diagrams.
- **Milestone 3 (Week 6, 05/05/2025)**: Develop core functionality (authentication, club selection, attendance tracking).
- **Milestone 4 (Week 8, 19/05/2025)**: Implement reminders, dashboard, and report generation.
- **Milestone 5 (Week 10, 02/06/2025)**: Add calendar and CSV storage features.
- **Milestone 6 (Week 12, 16/06/2025)**: Complete unit and integration testing.
- **Milestone 7 (Week 14, 30/06/2025)**: Conduct UAT and gather feedback.
- **Milestone 8 (Week 16, 03/07/2025)**: Finalize system, deploy, and obtain stakeholder sign-off.

---

13. Glossary

- **Actor**: A user or external entity interacting with the system (e.g., Student, Club Leader).
- **CSV**: Comma-Separated Values, a file format for storing tabular data.
- **GUI**: Graphical User Interface, a visual way for users to interact with the system.
- **Role-Based Access**: Security mechanism restricting access based on user roles.
- **UAT**: User Acceptance Testing, the process of verifying the system meets user needs.
- **Compass**: School management system used by GWSC for administrative tasks.

- **Inactive Member**: A club member who has missed 3 or more meetings.

---

14. Appendices

14.1 Interview Transcripts (Simulated)

- **GWSC Committee (03/04/2025)**:
  - **Question**: What features do you need to monitor club activities?
  - **Response**: We need a dashboard to monitor all club activities in real-time.
  - **Follow-Up**: Would a dashboard with real-time data meet your needs?
  - **Response**: Yes, it should also show membership numbers and attendance trends.
  - **Question**: Are there performance requirements for the system?
  - **Response**: The system must load within 2 seconds, even during peak usage.
- **Club Leader (Chess Club, 04/04/2025)**:
  - **Question**: How long does attendance tracking take?
  - **Response**: Attendance tracking takes too long—10–15 minutes per meeting.
  - **Follow-Up**: Would checkboxes in a GUI help reduce this time?
  - **Response**: Yes, I'd like checkboxes to mark attendance quickly.
  - **Question**: Do you face issues with students attending meetings?
  - **Response**: Yes, reminders would help ensure students show up on time.
- **Student (Year 10, 04/04/2025)**:
  - **Question**: Do you face any challenges with club meetings?
  - **Response**: I often forget club meetings because there's no reminder.
  - **Follow-Up**: Would in-app or email reminders help?
  - **Response**: Yes, that would be great.
  - **Question**: How do you find using Compass?
  - **Response**: Compass is hard to use on school computers—it's slow and confusing.
  - **Follow-Up**: What would make club management easier for you?
  - **Response**: I want an easy way to join clubs, like a list I can check off.
- **Admin (05/04/2025)**:
  - **Question**: What are your requirements for data integration?
  - **Response**: The system must export data to CSV for Compass integration.
  - **Follow-Up**: Are there specific formats required for the CSV?
  - **Response**: It should match Compass's import format—student ID, club ID, etc.
  - **Question**: Are there security concerns?
  - **Response**: We need role-based access to protect student data.
  - **Question**: What are the hardware constraints?
  - **Response**: School computers are old, so the system must be lightweight.

14.2 Survey Results (Simulated)

- **Question 1**: Do you want reminders for club meetings? (Students, n=50)
  - Yes: 85% (42/50)
  - No: 15% (8/50)
- **Question 2**: How time-consuming is attendance tracking? (Club Leaders, n=10)
  - 4/5 (Very Time-Consuming): 70% (7/10)
  - 3/5: 20% (2/10)
  - 2/5: 10% (1/10)
- **Question 3**: Do you prefer a graphical or text-based interface? (All Users, n=60)
  - Graphical: 90% (54/60)
  - Text-Based: 10% (6/60)
- **Question 4**: How often do you check club updates? (Students, n=50)
  - Weekly: 60% (30/50)
  - Daily: 20% (10/50)
  - Rarely: 20% (10/50)

14.3 Observation Notes

- **Chess Club Meeting (03/04/2025)**:
  - Leader spent 12 minutes recording attendance on paper.
  - 3 students arrived late, citing lack of reminders.
  - No centralized sign-up process; students asked the leader directly.
- **Drama Club Meeting (04/04/2025)**:
  - Leader spent 15 minutes on attendance due to a large group (25 students).
  - 5 students missed the meeting entirely.
  - Students struggled to use Compass on school computers (slow load times, navigation issues).

14.4 Report Analysis Summary

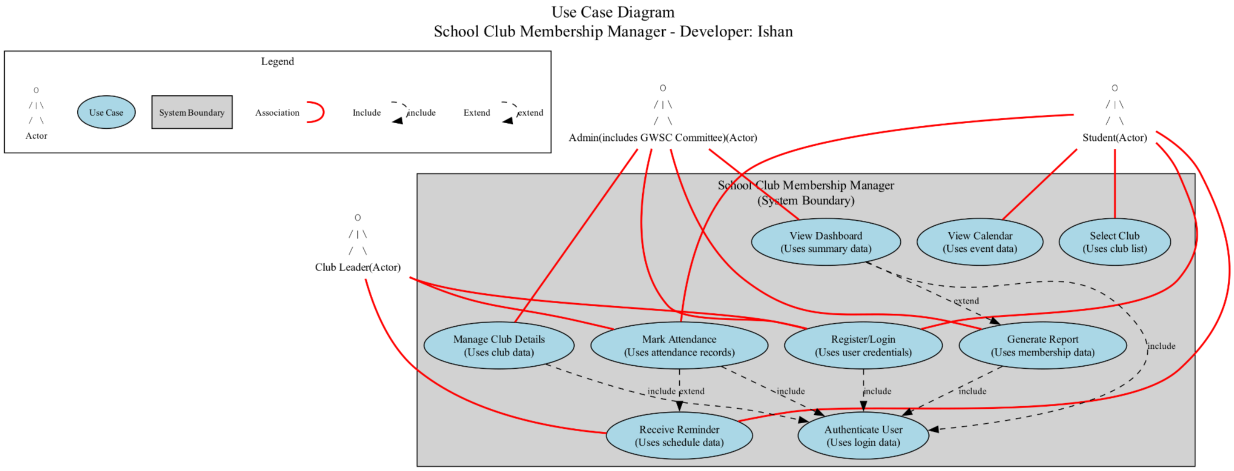
- **Membership Data**: Inconsistent formats across clubs (Excel, paper, Google Sheets).
- **Attendance Trends**: 20% of members are inactive (missed 3+ meetings).
- **Compass Limitations**: Lacks club-specific features; data entry is manual and error-prone.

14.5 System Diagrams

14.5.1 Use Case Diagram

Figure 1: Use Case Diagram (Version 1.0, Created: 21/04/2025)
Description: The Use Case Diagram illustrates the interactions between actors (Students, Club Leaders, Admins/GWSC Committee) and the system (School Club Membership Manager). It shows the main functionalities (use cases) and their relationships.

Diagram:



Elements:

- **Actors** (Orange Stick Figures):
  - Students
  - Club Leaders
  - Admins (includes GWSC Committee)
- **Use Cases** (Light Blue Ellipses):
  - Register/Login (uses user credentials)
  - Select Club (uses club list)
  - Mark Attendance (uses attendance records)
  - Receive Reminder (uses schedule data)
  - View Dashboard (uses summary data)
  - Generate Report (uses membership data)
  - View Calendar (uses event data)
  - Authenticate User (uses login data)
  - Manage Club Details (uses club data)
- **Relationships**:
  - **Associations**: Red bold lines (no arrowheads) between actors and use cases (e.g., Student → Select Club).
  - **Includes**: Black dashed arrows labeled "include" (e.g., Register/Login includes Authenticate User).
  - **Extends**: Black dashed arrows labeled "extend" (e.g., Mark Attendance extends to Receive Reminder if a student misses a meeting).
- **System Boundary**: Light grey box encapsulating all use cases within the School Club Membership Manager.
- **Legend**: Top-left corner, showing:
  - Actor (Orange Stick Figure)
  - Use Case (Light Blue Ellipse)
  - System Boundary (Light Grey Box)
  - Actor Interaction (Red Arrow)

  Notes on Limitations:

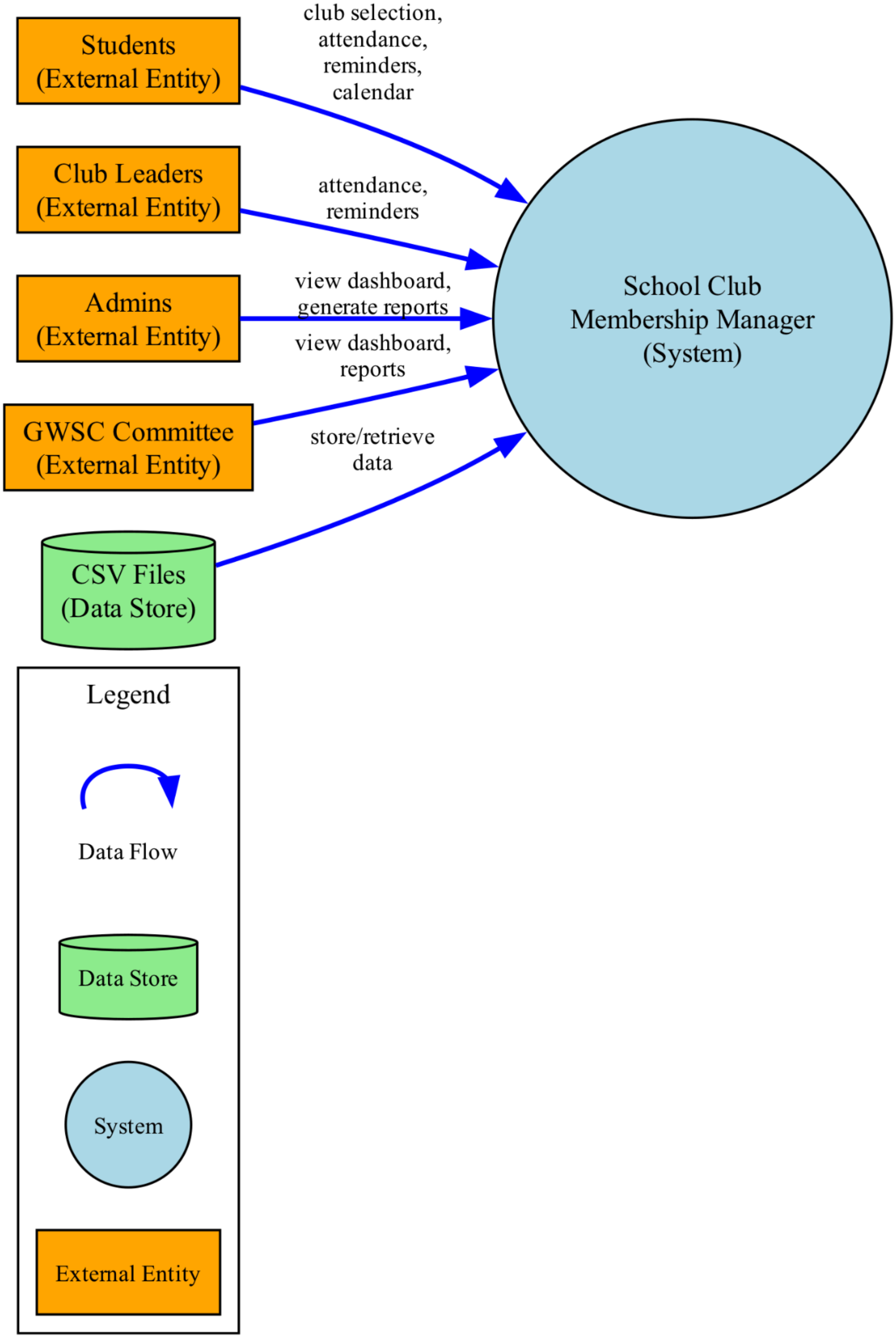
- Due to Graphviz 12.2.1 rendering issues, the UML standard notation `<<include>>` and `<<extend>>` could not be used. Instead, plain text labels "include" and "extend" are used.
- A NOTE node summarizing relationships was omitted due to Graphviz crashes with HTML-like labels.

14.5.2 Context Diagram

Figure 2: Context Diagram (Version 1.0, Created: 21/04/2025)
Description: The Context Diagram provides a high-level view of the School Club Membership Manager as a single process, showing its interactions with external entities and data stores.
Diagram:

Context Diagram

School Club Membership Manager - Developer: Ishan

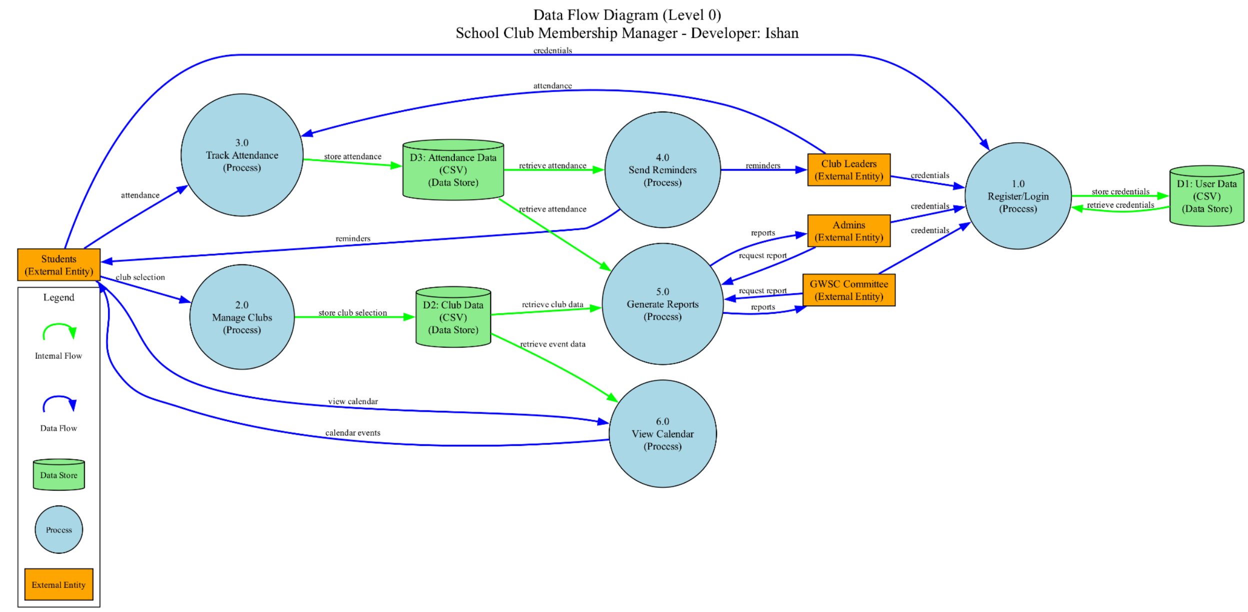


Elements:

- **System**: School Club Membership Manager (light blue circle).
- **External Entities** (orange rectangles):
  - Students
  - Club Leaders
  - Admins
  - GWSC Committee
- **Data Store** (light green cylinder):
  - CSV Files
- **Data Flows** (blue arrows):
  - Students ↔ System: club selection, attendance, reminders, calendar
  - Club Leaders ↔ System: attendance, reminders
  - Admins ↔ System: view dashboard, generate reports
  - GWSC Committee ↔ System: monitor activities, reports
  - CSV Files ↔ System: store/retrieve data
- **Legend**: Top-left corner, showing:
  - External Entity (Orange Rectangle)
  - System (Light Blue Circle)
  - Data Store (Light Green Cylinder)
  - Data Flow (Blue Arrow)

14.5.3 Data Flow Diagram (Level 0)

Figure 3: Data Flow Diagram (Level 0) (Version 1.0, Created: 21/04/2025)
Description: The Data Flow Diagram (Level 0) breaks down the School Club Membership Manager into major processes, showing how data flows between external entities, processes, and data stores.
Diagram:



Elements:

- **External Entities** (orange rectangles):
  - Students
  - Club Leaders
  - Admins
- **Processes** (light blue circles):
  - 1.0 Register/Login
  - 2.0 Manage Clubs
  - 3.0 Track Attendance
  - 4.0 Send Reminders
  - 5.0 Generate Reports
- **Data Stores** (light green cylinders):
  - D1: User Data (CSV)
  - D2: Club Data (CSV)
  - D3: Attendance Data (CSV)
- **Data Flows**:
  - **External Data Flows** (blue arrows):
    - Students → 1.0 Register/Login: credentials
    - Students → 2.0 Manage Clubs: club selection
    - Students → 3.0 Track Attendance: attendance
    - Club Leaders → 3.0 Track Attendance: attendance
    - Admins → 5.0 Generate Reports: request report
    - 4.0 Send Reminders → Students: reminders
    - 4.0 Send Reminders → Club Leaders: reminders
    - 5.0 Generate Reports → Admins: reports
  - **Internal Data Flows** (green arrows):
    - 1.0 Register/Login ↔ D1: store/retrieve credentials

        - 2.0 Manage Clubs → D2: store club selection
        - 3.0 Track Attendance → D3: store attendance
        - D3 → 4.0 Send Reminders: retrieve attendance
        - D2, D3 → 5.0 Generate Reports: retrieve club/attendance data
- **Legend**: Top-left corner, showing:
    - External Entity (Orange Rectangle)
    - Process (Light Blue Circle)
    - Data Store (Light Green Cylinder)
    - Data Flow (Blue Arrow)
    - Internal Flow (Green Arrow)

---

15. Traceability Matrix

The traceability matrix maps requirements to stakeholder needs and data sources, ensuring all needs are addressed.

Requirement	Stakeholder Need	Data Source
FR1.4 (Role-Based Access)	Protect student data	Admin Interview
FR4.1 (Automated Reminders)	Students forget meetings	Surveys (85% want reminders)
FR3.1 (Attendance Tracking)	Quick attendance tracking for leaders	Observations (10–15 mins)
FR5.1 (Admin Dashboard)	Oversight of club activities	GWSC Committee Interview
NFR1.1 (Intuitive GUI)	Easier navigation for students	Surveys (90% prefer GUI)

NFR2.1 (Load Time < 2s)	Handle peak usage	Admin Interview

---

16. Conclusion

This SRS provides a comprehensive blueprint for the **School Club Membership Manager**, addressing the needs of all stakeholders (GWSC Committee, club leaders, students, admin) while adhering to VCAA requirements for Unit 3, Outcome 2. The system will streamline club management, improve communication, and enhance data consistency, all within the given constraints of timeline, budget, and hardware. With rigorous testing, risk mitigation, and change management processes in place, the project is well-positioned for successful deployment by 03/07/2025.

---

17. Submission Notes for VCAA

- **Alignment with U3O2**: This SRS includes detailed data collection and analysis (Section 2), system diagrams (Appendix 14.5), and a comprehensive requirements specification (Sections 3–4), meeting VCAA expectations for analytical and design documentation.
- **Diagrams**: Embedded information (titles, legends, labels) ensures clarity without the need for separate documentation.
- **Appendices**: Provide raw data to support the analysis, demonstrating a thorough approach to requirements gathering.