SC2006-Software Engineering

Lab 1 Deliverables

Group Member	**Matric Number**
Lim Kiat Yang Ryan	U2421937D
Yu Wenhao	U2421425F
Yong Chee Seng	U2420563K
Peng Sizhe	U2423895H
Tarun Ilangovan	U2422251A

A. Documentation of functional and non-functional requirements

A.1 Functional Requirements

1. The user shall be able to query the system for schools using zero, one, or multiple search requirements.
    1.1. If no search requirements are provided, the system shall include all schools in the results.
    1.2. If one or more search requirements are provided, the system shall only include schools that satisfy all search requirements in the results.
    1.3. The user shall be able to filter schools by name.
        1.3.1. The name input shall be a text string of 0–512 characters.
        1.3.2. The name input shall be a string of 0 characters by default.
        1.3.3. If 0 characters are provided (empty input), the query shall not filter by name.
        1.3.4. If 1 or more characters are provided, the system shall include only schools whose name contains the input substring (case-insensitive) in the results.
    1.4. The user shall be able to filter schools by cut-off point range.
        1.4.1. The cut-off point range shall consist minimum cut-off point and maximum cut-off point.
        1.4.2. The minimum cut-off point input shall be an integer within 4-32.
            1.4.2.1. The minimum cut-off point input shall be 4 by default.
        1.4.3. The maximum cut-off point input shall be an integer within 4-32.
            1.4.3.1. The maximum cut-off point input shall be larger than or equal to the minimum cut-off point.
            1.4.3.2. The maximum cut-off point input shall be 32 by default.
        1.4.4. The system shall include only schools whose cut-off points are within minimum and maximum cut-off point (inclusive) in the results.
    1.5. The user shall be able to filter schools based on their location.
        1.5.1. The location input shall be a multi-choice dropdown with all locations available.
        1.5.2. The location input shall be empty (no selected locations) by default.
        1.5.3. If 0 locations are provided, the query shall not filter by location.

1.5.4. If 1 or more locations are provided, the system shall include only schools whose location is in the selected locations in the results.

1.6. The user shall be able to filter schools based on their Co-Curricular Activities (CCA) offered.

1.6.1. The CCA input shall be a multi-choice dropdown with all CCA available.

1.6.2. The CCA input shall be empty (no selected CCA) by default.

1.6.3. If 0 CCA are provided, the query shall not filter by CCA.

1.6.4. If 1 or more CCA are provided, the system shall include only schools whose 1 or more CCAs offered are in the selected CCA in the results.

1.7. The user shall be able to filter schools based on their subjects offered.

1.7.1. The subjects input shall be a multi-choice dropdown with all subjects available.

1.7.2. The subjects input shall be empty (no selected subjects) by default.

1.7.3. If 0 subjects are provided, the query shall not filter by subjects.

1.7.4. If 1 or more subjects are provided, the system shall include only schools whose 1 or more subjects offered are in the selected subjects in the results.

1.8. The user shall be able to filter schools based on their level.

1.8.1. The level input shall be a single-choice dropdown with all levels available and the option “All”.

1.8.2. The level input shall be “All” by default.

1.8.3. If “All” is selected, the query shall not filter by level.

1.8.4. If levels other than “All” are selected, the system shall include only schools whose level is the selected level in the results.

1.9. The user shall be able to filter schools based on their nature code.

1.9.1. The nature code input shall be a single-choice dropdown with all nature codes available and the option “All”.

1.9.2. The nature code input shall be “All” by default.

1.9.3. If “All” is selected, the query shall not filter by nature.

1.9.4. If nature codes other than “All” are selected, the system shall include only schools whose nature code is the selected nature code in the results.

1.10. The user shall be able to filter schools based on their type.

1.10.1. The type input shall be a single-choice dropdown with all types available and the option “All”.

1.10.2. The type input shall be “All” by default.

1.10.3. If “All” is selected, the query shall not filter by type.

1.10.4. If types other than “All” are selected, the system shall include only schools whose type is the selected type in the results.

2. The user shall be able to view school details when selecting a school from the query results.

2.1. The system shall display the general information of school:

2.1.1. The system shall display the school’s name.

2.1.2. The system shall display the school’s level.

2.1.3. The system shall display the school’s type.

2.1.4. The system shall display the school’s nature code.

2.1.5. The system shall display the school’s session code.

2.1.6. The system shall display the school’s cut-off point range.

2.2. The system shall display geographical information of the school:

2.2.1. The system shall display the school’s address.

2.2.2. The system shall display the school’s postal code.

2.2.3. The system shall display the school’s location.

2.2.4. The system shall display the school’s nearby MRT station(s).

2.2.5. The system shall display the school’s nearby bus station(s).

2.2.6. The system shall display an interactive map interface (e.g., Google Maps or equivalent) showing the school’s location.

2.3. The system shall display the contact information of the school:

2.3.1. The system shall display the school’s official website as a hyperlink.

2.3.2. The system shall display the school’s email address as a hyperlink that opens the user’s default mail client.

2.3.3. The system shall display the school’s phone number as a hyperlink that initiates a call on supported devices.

2.3.4. The system shall display the school’s fax number.

2.4. The system shall display the subjects offered by the school.

2.4.1. The system shall display subjects in ascending lexicographical order.

2.4.2. The system shall display only the first 5 subjects by default.

2.4.3. The system shall display all subjects upon user request.

2.5. The system shall display the CCAs offered by school.
    2.5.1. The system shall display CCAs in ascending lexicographical order.
    2.5.2. The system shall display only the first 5 CCAs by default.
    2.5.3. The system shall display all CCAs upon user request.
3. The user shall be able to compare 2 schools side by side from the query results.
    3.1. The system shall display for both schools all information specified in requirement 2.
4. The system shall enforce authentication for all functions that require user identity
    4.1. The user shall be able to create an account.
        4.1.1. The user shall provide a username.
            4.1.1.1. The username shall be a string of 6-14 characters.
            4.1.1.2. The username shall be unique across the user database.
        4.1.2. The user shall provide a password.
            4.1.2.1. The password shall be a string of at least 8 characters.
            4.1.2.2. The password shall meet minimum complexity requirements:
                4.1.2.2.1. The password shall have at least 1 lowercase letter.
                4.1.2.2.2. The password shall have at least 1 uppercase alphabet.
                4.1.2.2.3. The password shall have at least 1 number.
                4.1.2.2.4. The password shall have at least 1 non-alphanumeric character.
    4.2. The user shall be able to log into their account.
        4.2.1. The user shall be able to log into their account with valid credentials (username and password).
    4.3. The system shall save user sessions using a secure method after successful login.
        4.3.1. The system shall automatically invalidate user sessions no later than 7 days after the last successful login.
    4.4. The user shall be able to log out of their account.
        4.4.1. The system shall invalidate user sessions after logout.
5. The system shall have a comment section under the school detail page.
    5.1. The system shall display all comments related to the specific school.
        5.1.1. The system shall display comments in descending chronological order.
        5.1.2. The system shall display at most 2 replies for each comment by default.

5.1.3. The system shall display at all replies for each comment upon request.

5.1.4. The system shall display upvote count for each comment.

5.1.5. The system shall display downvote count for each comment.

5.2. The user shall be able to create comments.

5.2.1. The system shall enforce authentication to create comments.

5.2.2. Each comment shall be associated with a specific school.

5.2.3. The comment shall be a string with at most 1024 characters.

5.3. The user shall be able to reply to comments.

5.3.1. The system shall enforce authentication to reply to comments.

5.3.2. Each reply shall be associated with a specific comment.

5.3.3. The reply shall be a string with at most 1024 characters.

5.4. The user shall be able to vote for comments.

5.4.1. The system shall enforce authentication to vote for comments.

5.4.2. The user shall be able to upvote a comment.

5.4.2.1. If the user has not previously voted, an upvote shall be added.

5.4.2.2. If the user has already upvoted the comment, the upvote shall be removed (toggle behavior).

5.4.2.3. If the user had already previously downvoted the comment, the downvote shall be removed and an upvote shall be added.

5.4.3. The user shall be able to downvote a comment.

5.4.3.1. If the user has not previously voted, a downvote shall be added.

5.4.3.2. If the user has already downvoted the comment, the downvote shall be removed (toggle behavior).

5.4.3.3. If the user had already previously upvoted the comment, the upvote shall be removed and a downvote shall be added.

6. The user shall be able to save a school as favourite.

6.1. The system shall enforce authentication to save as favourite.

6.2. If the user has not previously saved the school as a favourite, the system shall add the school to the user's favourites.

6.3. If the user has already saved the school as a favourite, the system shall remove the school from the user's favourites.

6.4. The user shall be able to view a list of their saved schools on a dedicated Favourites page.

A.2 Non-Functional

Usability	1. System shall be intuitive to users with minimal technical knowledge 1.1. 80% of users shall be able to make a simple school search within 2 minutes of using the website 1.1.1. A simple school search will encompass a simple name search (e.g. XXX secondary school) or a single-field filter (e.g. filter by zone code) 1.2. System UI shall adhere to Nielsen Norman group's usability heuristics, with minimalistic design, clear labels and consistent layout 2. System shall have a responsive design compatible with desktop, tablet and mobile displays 3. System shall provide accessibility support in compliance with WCAG 2.1 AA guidelines 3.1. System shall focus on screen reader compatibility and keyboard navigation support 4. System shall provide helpful validation messages and error hints on erroneous user inputs 5. System shall provide feedback (e.g., loading indicators or transition pages) for operations exceeding 1 second, to inform users that the system is processing the request.
Reliability	1. System shall be available 99.5% of the time, excluding periods of planned maintenance 2. System shall display a fallback maintenance page in the event of downtime 3. System shall schedule planned maintenance during periods of low web traffic, between 1:00 AM and 6:00 AM SGT
Performance	1. System shall respond to school query requests within 2 seconds for 95% of queries under standard load conditions.

	1.1. System shall use optimization techniques (e.g., caching) for school query requests. 2. System shall respond to database queries within 5 seconds for 95% of operations under standard load conditions. 3. System shall support at least 100 concurrent users with no more than 10% increase in average response time compared to single-user load.
Persistence	1. System shall persist only necessary information via client-side storage (e.g., cookies, localStorage). 1.1. System shall invalidate client-side cookies no later than 7 days after creation. 2. System shall persist user-specific data for authenticated users in the server-side database. 2.1. System shall protect all persisted information against unauthorized access and tampering.
Security	1. System shall use HTTPS with TLS 1.2 or higher to encrypt all communications between client and server. 2. System shall sanitize and validate all user inputs to prevent security vulnerabilities such as XSS, SQL injection, CSRF, and command injection. 3. System shall implement rate limiting and/or CAPTCHA to mitigate automated bot attacks. 4. System shall implement throttling, monitoring, and alerts to mitigate and detect DoS/DDoS attacks.
Authentication	1. System shall protect session information against unauthorized access and tampering using secure mechanisms (e.g., HTTP-only Secure cookies, signed tokens). 2. System shall implement protections against brute-force login attempts (e.g., rate limiting, captcha, temporary lockout). 3. System shall hash and salt all passwords using an industry-standard algorithm (e.g., bcrypt, Argon2) before storage.

Data Storage	1. System shall abstract data access so that the underlying database technology can be replaced with any database system, provided it supports the required data operations (e.g., CRUD, queries, indexing).
Data Collection	1. System shall refresh school data from the Singapore Open Government Dataset via the data.gov.sg API at least once per week. 2. System shall update the cut-off point data within 7 days of the data becoming available. 3. System shall cross-compare and validate newly fetched data to ensure consistency and accuracy before updating the database.
Data Protection and Privacy	1. System and its maintaining organization shall comply with Singapore PDPA regulations for the storage and handling of personal data. 2. System shall only collect the minimum necessary data required for functionality. 3. System shall implement mechanisms to enforce data retention limits in accordance with PDPA. 4. System shall provide a clear and accessible privacy policy describing: 4.1. What data is collected 4.2. Why it is collected 4.3. How long it will be retained 4.4. How it will be used 5. System shall only collect personal data with the user's explicit consent. 5.1. System shall require the user to agree to the privacy policy when creating an account. 6. System shall allow users to request removal of their personal data by contacting the maintaining party.

Terms and Conditions	1. System shall provide a clear and accessible Terms and Conditions describing: 1.1. Actions that are considered violations. 1.2. Consequences of such violations (e.g., warnings, account suspension, or deletion). 2. System shall require users to explicitly agree to the Terms and Conditions when creating an account. 3. System shall allow authorized administrators to delete, restrict, or suspend user accounts that violate the Terms and Conditions.
Logging	1. System shall maintain logs for user actions and application errors. 2. System shall ensure that sensitive information (e.g., passwords, session tokens) is never logged in plain text. 3. System shall restrict access to logs to authorized personnel only.
Maintenability	1. System shall support unit testing and integration testing 1.1. Unit test coverage should be >80% 1.2. System shall undergo user acceptance testing (UAT) before release with >10 representative users 1.2.1. UAT shall pass with >90% task completion and >80% user satisfaction rating

B. Data Dictionary

B.1 School

Term	Description
School	An educational institution in Singapore. The system's list of schools and their information shall be sourced from the Singapore Open Government Dataset: https://data.gov.sg/
Search Criteria	The fields used to filter schools in the system. Includes: • Name • Cut-off Point Range • Location • CCA • Subjects • Level • Nature Code • Type
Name	The official name of the school; can be used as a school identifier.
Cut-off Point	A student's PSLE score used to determine the posting group and indicative level for subjects. Range: 4–32 (4 = best, 32 = worst).
Location	The DGP (Development Guide Plan) code of the school, such as Woodlands or Jurong West. Represents the planning area as defined by the Urban Redevelopment Authority.
CCA	Co-Curricular Activities offered by the school.
Subjects	Academic subjects offered by the school.
Level	Education level of the school (e.g., Primary, Secondary, Junior College).
Nature Code	Type of school nature (e.g., Co-ed, Boys, and Girls).
Type	Type of school (e.g., Government, Government-aided, etc.).

Session Code	The academic session type (e.g., Single session and Full day).
Address	Full street address of the school.
Postal Code	Postal code for the school address.
Nearby MRT Station	MRT stations that are close to the school.
Nearby Bus Station	Bus stops that are close to the school.
Website	URL of the school's official website.
Email	Official school email address.
Phone Number	Official contact phone number for the school.
Fax Number	Official fax number for the school

B.2 User

Term	**Description**
User ID	Unique identifier for each user.
Username	Unique username chosen by the user.
Password	User's password.
Last Login	Timestamp of the user's last successful login.

B.3 Comment

Term	**Description**
Comment ID	Unique identifier for each comment.
User ID	Identifier of the user who created the comment.

School ID	Identifier of the school associated with the comment.
Content	Content of the comment.
Upvote Count	Number of upvotes for the comment. Updated whenever a user upvotes or removes an upvote.
Downvote Count	Number of downvotes for the comment. Updated whenever a user downvotes or removes a downvote.
Created At	Timestamp the comment was created.

B.4 Reply

Term	**Description**
Reply ID	Unique identifier for each reply.
User ID	Identifier of the user who created the reply.
Comment ID	Identifier of the parent comment.
Content	Content of the reply.
Created At	Timestamp the reply was created.

B.5 Vote

Term	**Description**
Upvote	Users like or agree with the comment.
Downvote	Users dislike or disagree with the comment.
Vote ID	Unique identifier for each vote.
Comment ID	Identifier of the comment being voted.
User ID	Identifier of the user who created the vote.

Type	Upvote or downvote.

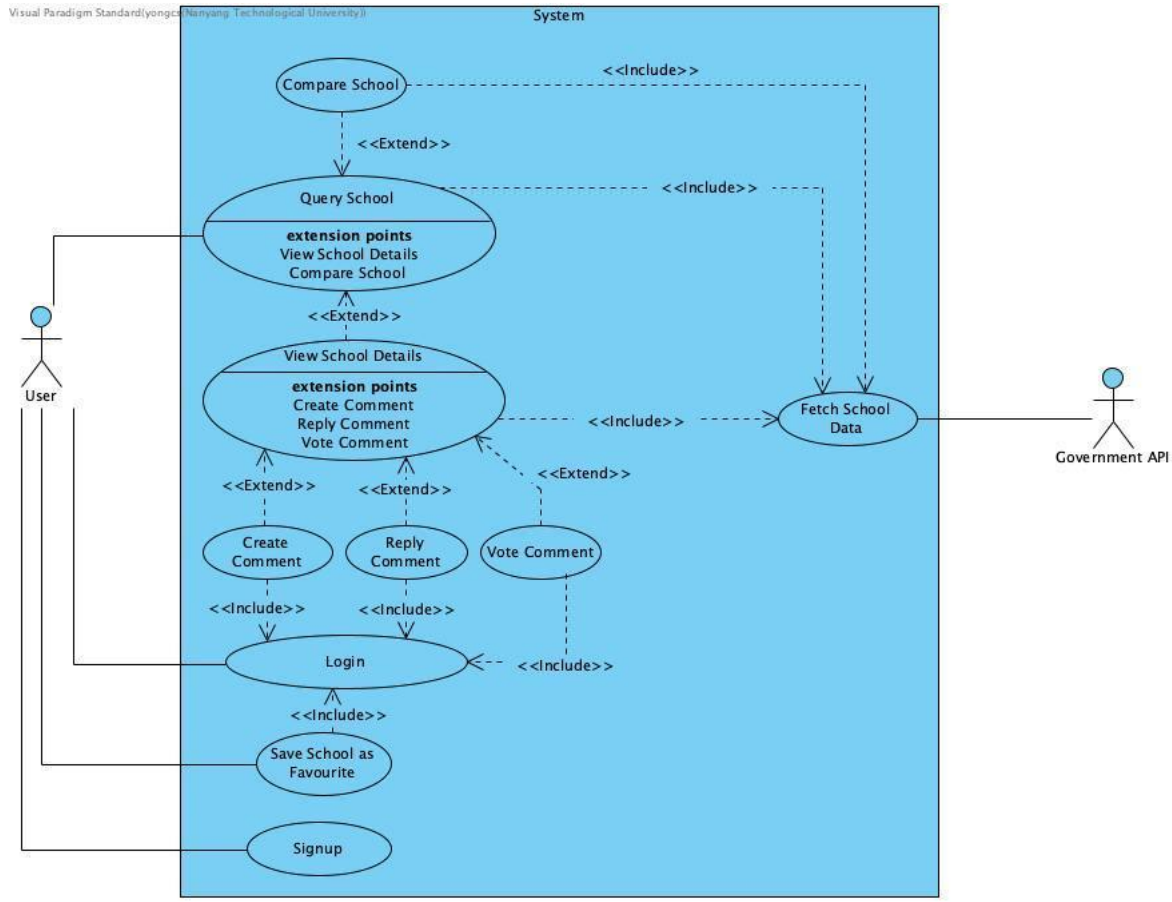
B.6 Favourite School

Term	**Description**
User ID	Identifier of the user who saved the school.
School ID	Identifier of the saved school.

B.6 Actor

Term	**Description**
Government API	The API exposed by Singapore Open Government Dataset: https://data.gov.sg/.

Use Case Diagram:



C. Use Cases

1.1. UCQ

This section describes all use cases related to querying and retrieving a list of results. All use case IDs in this section are prefixed with **UCQ**.

1.1.1 UCQ-1 Query School

Use Case ID:	UCQ-1		
Use Case Name:	Query School		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to query school based on different criteria, and system shall show list of schools that satisfy all the criteria.
Preconditions:	NA
Postconditions:	1. System displays a list of schools that match all selected criteria or shows “no results” if none found.
Priority:	High
Frequency of Use:	Estimated 50–200 per day across all users
Flow of Events:	1. User navigates to “Query School” page. 2. System initiates **UCSY-1 Fetch School Data from Government API** to retrieve the latest school dataset. 3. System displays the full list of schools. 4. System displays available query criteria (e.g., name, region, subject, CCA). 5. User enters one or more criteria. 6. User submits the query. 7. System validates the query input. 8. System filters the school data according to criteria. 9. System displays a list of matching schools with summary information.
Alternative Flows:	AFS3: **UCSY-1 Throws Exception**

	1. System displays error “Unexpected error occurred. Please try again”. AFS8: **No Criteria Entered** 1. System continues showing all schools. AFS9: **No Results Found** 1. If no school satisfies all the criteria, system displays message “No result found”.
Exceptions:	NA
Includes:	1. UCSY-1 Fetch School Data from Government API
Special Requirements:	1. System shall allow selection of criteria from predefined options where applicable (e.g., when filtering by subject, the system shall provide a list of available subjects instead of requiring manual text input) 2. System shall complete the filter within 2 seconds under normal load condition.
Assumptions:	1. The **“Query School”** page is the main entry point of the application.
Notes and Issues:	TBD: 1. Shall the system allow **human text input** (e.g., “a school near Area A offering subjects B and C”)? a. To be resolved by: Yu Wen Hao b. Due date: 17 September 2025

1.2. UCD

This section describes all use cases related to retrieving and displaying details of a result. All use case IDs in this section are prefixed with **UCD**.

1.2.1 UCD-1 View School Details

Use Case ID:	UCD-1		
Use Case Name:	View School Details		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to view the details of a selected school, and system shall display the full information of that school.
Preconditions:	1. System has received the identifier of the selected school (e.g., school name or unique ID).
Postconditions:	1. System displays the full details of the selected school. 2. System displays any available user comments associated with the school.
Priority:	High
Frequency of Use:	Estimated 100–500 per day across all users
Flow of Events:	1. User selects a school from the query results (or other entry points). 2. System initiates **UCSY-1 Fetch School Data from Government API** to retrieve the details of the selected school using its identifier. 3. System displays the full details of the selected school. 4. System retrieves users' comments related to the school. 5. System displays the retrieved user comments.
Alternative Flows:	AFS3a: **UCSY-1 Throws Exception** 1. System displays error "Unexpected error occurred. Please try again". AFS3b: **School Not Found**

	1. If school with the identifier is not found, system displays error “School not found”. AFS5: **Comments Unavailable** 1. If comments cannot be retrieved, system displays error “Comments unavailable at the moment”.
Exceptions:	NA
Includes:	1. UCSY-1 Fetch School Data from Government API
Special Requirements:	1. System shall show the details within 2 seconds under normal load condition.
Assumptions:	NA
Notes and Issues:	NA

1.2.2 UCD-2 Compare School

Use Case ID:	UCD-2		
Use Case Name:	Compare School		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to view and compare the details of 2 selected schools side by side.
Preconditions:	1. System has received the identifier of the 2 selected schools (e.g., school name or unique ID).
Postconditions:	1. System displays the full details of the 2 selected schools side by side. 2. System displays any available user comments associated with the 2 selected schools.
Priority:	Medium
Frequency of Use:	Estimated 10–50 per day across all users
Flow of Events:	1. User selects 2 schools from the query results (or other entry points). 2. System initiates **UCSY-1 Fetch School Data from Government API** to retrieve the details of the 2 selected schools using their identifier. 3. System displays the full details of the 2 selected schools side by side. 4. System retrieves users' comments related to the 2 selected schools. 5. System displays the retrieved user comments.
Alternative Flows:	AFS3a: **UCSY-1 Throws Exception** 2. System displays error "Unexpected error occurred. Please try again". AFS3b: **School Not Found** 2. If school with the identifier is not found, system displays error "School not found".

	AFS5: **Comments Unavailable** 2. If comments cannot be retrieved, system displays error "Comments unavailable at the moment".
Exceptions:	NA
Includes:	1. UCSY-1 Fetch School Data from Government API
Special Requirements:	1. System shall show the details within 2 seconds under normal load condition. 2. System shall reuse the UI components of **UCD-1 View School Details** where applicable to ensure consistency and maintainability. 3. System shall present the comparison of both schools in a responsive layout that avoids UI overlay issues or excessive horizontal scrolling across all supported devices.
Assumptions:	NA
Notes and Issues:	NA

1.3. UCA

This section describes all use cases related to authentication. All use case IDs in this section are prefixed with **UCA**.

1.3.1 UCA-1 Login

Use Case ID:	UCA-1		
Use Case Name:	Login		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to login to their account by providing valid credentials.
Preconditions:	1. User has signed up an account.
Postconditions:	1. System recognizes the user as authenticated and grants access to logged-in user functionalities. 2. System displays the account username (or equivalent identifier) in the UI.
Priority:	Medium
Frequency of Use:	Estimated 5-10 per day across all users
Flow of Events:	1. User navigates to “Login” page. 2. User inputs username and password. 3. System retrieves the user account information corresponding to the entered username. 4. If user exists, system validates the entered password against the stored password. 5. If password is correct, system creates a session for the authenticated user. 6. System displays message “Login successfully”. 7. System redirects the user to the last page visited (or to the home page if no prior page exists).
Alternative Flows:	AFS4a: **User Not Found**

	1. If the entered username does not exist, system displays error “Invalid username”. AFS4b: **User Unavailable** 1. If system cannot access user data, it displays error “Login unavailable. Please try again later”. AFS5: **Incorrect Password** 1. If the password is incorrect, system displays error “Invalid password”.
Exceptions:	NA
Includes:	NA
Special Requirements:	1. System shall complete the login process within 5 seconds under normal load conditions. 2. System shall protect against brute-force login attempts (e.g., account lockout, reCAPTCHA).
Assumptions:	NA
Notes and Issues:	NA

1.3.2 UCA-2 Signup

Use Case ID:	UCA-2		
Use Case Name:	Signup		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to sign up an account by providing valid credentials.
Preconditions:	NA
Postconditions:	1. System stores user credentials securely. 2. System allow user to login with the provided credentials
Priority:	Medium
Frequency of Use:	Estimated 3-5 per day across all users
Flow of Events:	1. User navigates to “Signup” page. 2. User inputs username, password, and confirm password.

	3. System checks whether the username is already taken. 4. If the username is available, the system validates that the password meets the security policy (e.g., minimum length of 8 characters, includes uppercase, lowercase, number, and special symbol). 5. If password is secure, system verifies that the password and confirm password match. 6. If the confirm password is same as confirm password, system stores user credentials. 7. The system displays message “Account created. You can login now”. 8. System redirects the user to the “Login” page.
Alternative Flows:	AFS4a: **Username Taken** 1. If the entered username does not exist, system displays error “Username already exists. Please choose another”. AFS4b: **User Unavailable** 1. If system cannot access user data, system displays error “Signup unavailable. Please try again later”. AFS5: **Weak Password** 1. If the password does not meet the security policy, system displays error: “Password does not meet security requirements”. AFS6: **Password Mismatch** 1. If the password and confirm password do not match, system displays the error: “Passwords do not match”. AFS7: **System Error** 1. If credentials cannot be stored due to a system error, system displays error: “Signup unavailable. Please try again later”.
Exceptions:	NA
Includes:	NA

Special Requirements:	1. System shall enforce a strong password policy (e.g., minimum length of 8 characters, including uppercase, lowercase, numbers, and special symbols). 2. System shall complete the signup process within 5 seconds under normal load conditions. 3. System shall protect against automated/bot signups (e.g., reCAPTCHA or an equivalent). 4. System shall store all user passwords using industry-standard encryption methods (e.g., bcrypt). 5. The system shall ensure that users explicitly agree to the **Terms and Conditions** and **Privacy Policy** in a clear and visible manner before account creation.
Assumptions:	NA
Notes and Issues:	NA

1.4. UCS

This section describes all use cases related to creating, modifying, and storing data in the system. All use case IDs in this section are prefixed with **UCS**.

1.4.1 UCS-1 Create Comment

Use Case ID:	UCS-1		
Use Case Name:	Create Comment		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to create and submit a comment under any "School Details" page.
Preconditions:	1. User has logged in to an account. 2. User has navigated to a valid "School Details" page.
Postconditions:	1. System stores the submitted comment. 2. System displays the new comment under the related school's comments section.
Priority:	Low
Frequency of Use:	Estimated 3-5 per day across all users
Flow of Events:	1. User navigates to "School Details" page. 2. User inputs comment into the comment input field. 3. User submits the comment. 4. System validates the input (e.g., not empty, within allowed length). 5. If valid, system stores the comment. 6. System refreshes or updates the page (e.g., via AJAX or equivalent). 7. System displays the newly created comment in the comments section.
Alternative Flows:	AFS2: **User Not Logged In** 1. System displays message "You need to login to comment". AFS5: **Invalid Input**

	1. System displays a clear error message explaining why input is invalid (e.g., empty or too long). AFS6: **System Error** 1. If comment cannot be stored due to a system error, system displays error: “Comment unavailable. Please try again later”.
Exceptions:	NA
Includes:	UCA-1 Login
Special Requirements:	1. System shall complete the comment creation process within 5 seconds under normal load conditions. 2. System shall protect against automated/bot attack (e.g., reCAPTCHA or an equivalent). 3. System shall sanitize and validate user input to prevent malicious attacks (e.g., XSS, SQL injection).
Assumptions:	NA
Notes and Issues:	NA

1.4.2 UCS-2 Reply Comment

Use Case ID:	UCS-2		
Use Case Name:	Reply Comment		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to reply to a comment under any “School Details” page.
Preconditions:	1. User has logged in to an account. 2. User has navigated to a valid “School Details” page.
Postconditions:	1. System stores the submitted reply. 2. System displays the new reply under the related comments.
Priority:	Low
Frequency of Use:	Estimated 3-5 per day across all users
Flow of Events:	1. User navigates to “School Details” page. 2. User selects to reply to a comment.

	3. User inputs reply into the input field. 4. User submits the reply. 5. System validates the input (e.g., not empty, within allowed length). 6. If valid, system stores the reply. 7. System refreshes or updates the page (e.g., via AJAX or equivalent). 8. System displays the newly created reply under the related comment.
Alternative Flows:	AFS3: **User Not Logged In** 1. System displays message "You need to login to reply". AFS6: **Invalid Input** 1. System displays a clear error message explaining why input is invalid (e.g., empty or too long). AFS7: **System Error** 1. If reply cannot be stored due to a system error, system displays error: "Reply unavailable. Please try again later".
Exceptions:	NA
Includes:	UCA-1 Login
Special Requirements:	1. System shall complete the reply creation process within 5 seconds under normal load conditions. 2. System shall protect against automated/bot attack (e.g., reCAPTCHA or an equivalent). 3. System shall sanitize and validate user input to prevent malicious attacks (e.g., XSS, SQL injection).
Assumptions:	NA
Notes and Issues:	NA

1.4.3 UCS-3 Vote Comment

Use Case ID:	UCS-3		
Use Case Name:	Vote Comment		
Created By:	Yong Chee Seng	Last Updated By:	

Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to upvotes or downvotes a comment under any “School Details” page.
Preconditions:	1. User has logged in to an account. 2. User has navigated to a valid “School Details” page.
Postconditions:	1. System stores the submitted upvote or downvote. 2. System displays the new vote count under the related comment.
Priority:	Low
Frequency of Use:	Estimated 5-10 per day across all users
Flow of Events:	1. User navigates to “School Details” page. 2. User selects a comment to upvote or downvote. 3. System stores the vote. 4. System refreshes or updates the page (e.g., via AJAX or equivalent). 5. System displays the new upvote and downvote count under the related comments.
Alternative Flows:	AFS3a: **User Not Logged In** 1. System displays message “You need to login to upvote or downvote”. AFS3b: **User Already Voted** 1. If the user has already voted the comment, the system toggles the vote. AFS4: **System Error** 1. If reply cannot be stored due to a system error, system displays error: “Upvote and downvote unavailable. Please try again later”.
Exceptions:	NA
Includes:	UCA-1 Login

Special Requirements:	1. System shall complete the process within 5 seconds under normal load conditions. 2. System shall protect against automated/bot attack (e.g., reCAPTCHA or an equivalent).
Assumptions:	NA
Notes and Issues:	NA

1.4.4 UCS-4 Save School as Favourite

Use Case ID:	UCS-4		
Use Case Name:	Save School as Favourite		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	User (Initiating)
Description:	User shall be able to save a school as favourite.
Preconditions:	1. User has logged in to an account. 2. User has navigated to a valid entry point (e.g., Query School, School Details).
Postconditions:	1. System stores the school as favourite. 2. System visually indicates the favourite status of the school.
Priority:	Medium
Frequency of Use:	Estimated 5-10 per day across all users
Flow of Events:	1. User navigates to valid entry point. 2. User selects a school to save as favourite. 3. System stores the school as favourite for the user. 4. System displays a clear sign the school is saved as favourite (e.g., filled star).
Alternative Flows:	AFS3a: **User Not Logged In** 1. System displays message “You need to login to save the school”. AFS3b: **User Already Saved as Favourite** 1. If the user has already saved the school as favourite, the system removes it from favourites and updates the UI indicator. AFS4: **System Error** 1. If the school cannot be stored or removed due to a system error, system displays: “Unable to update favourite. Please try again later.”
Exceptions:	NA

Includes:	UCA-1 Login
Special Requirements:	1. System shall complete the process within 5 seconds under normal load conditions. 2. System shall protect against automated/bot attack (e.g., reCAPTCHA or an equivalent).
Assumptions:	NA
Notes and Issues:	NA

1.5. UCSY

This section describes all use cases related to system-level actions not directly initiated by the user. All use case IDs in this section are prefixed with **UCSY**.

1.5.1 UCSY-1 Fetch School Data from Government API

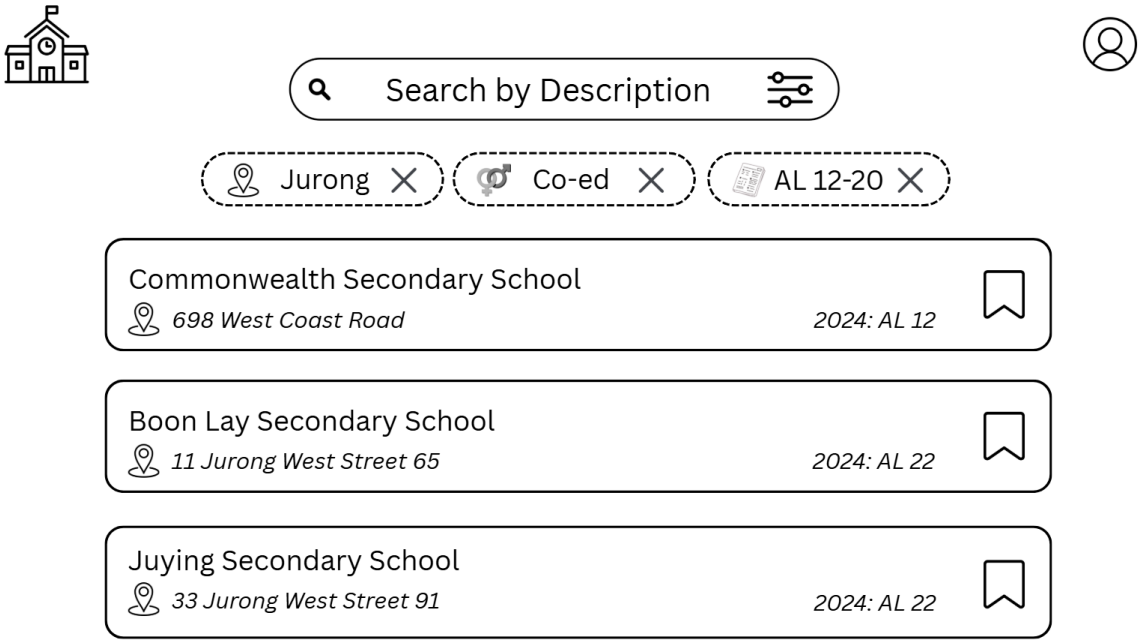
Use Case ID:	UCSY-1		
Use Case Name:	Fetch School Data from Government API		
Created By:	Yong Chee Seng	Last Updated By:	
Date Created:	4 September 2025	Date Last Updated:	

Actor:	Government API
Description:	System shall retrieve available school data from MOE API.
Preconditions:	NA
Postconditions:	1. System caches the retrieved school data. 2. System returns school data when requested.
Priority:	High
Frequency of Use:	Estimated 100-500 per day across all users
Flow of Events:	1. System initiates **UCSY-1 Fetch School Data from Government API.** 2. If school data is not cached, the system calls the MOE API to retrieve all school data. 3. Government API returns all school data. 4. System caches the retrieved school data. 5. If no school identifier is provided, the system returns all cached school data.
Alternative Flows:	AFS2: **School Data is Cached** 1. Continue with step 5. AFS5: **School Identifier is Provided** 1. System returns only the corresponding school data.
Exceptions:	EX1: **MOE API Unavailable** 1. System throw exception to be caught by initiating process.
Includes:	NA

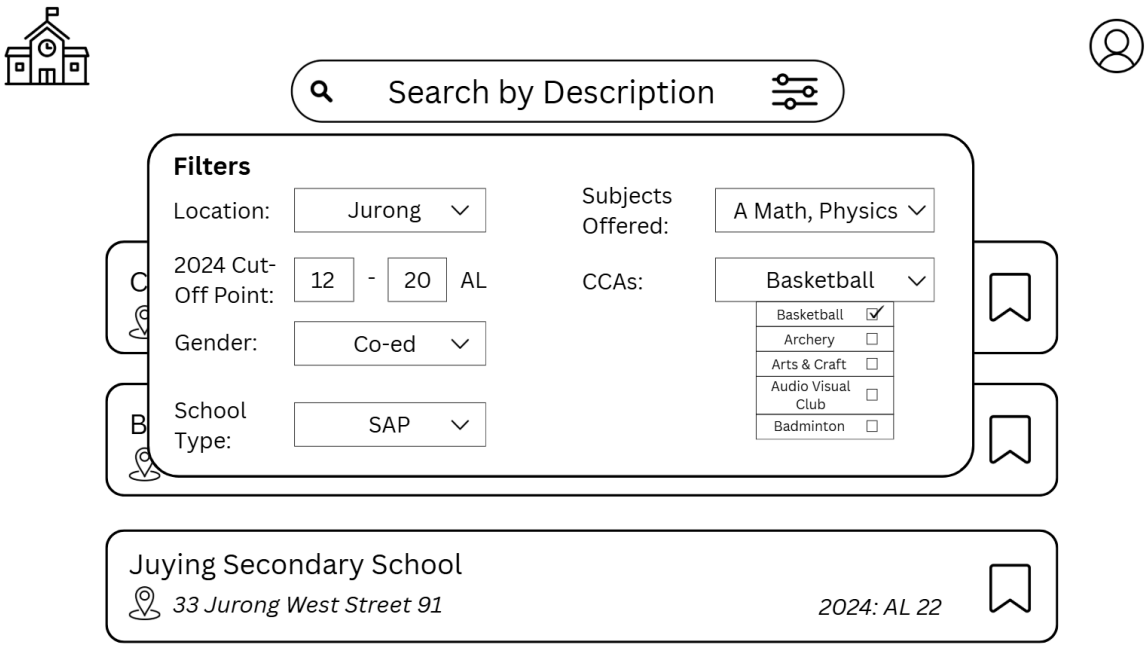
Special Requirements:	1. System shall complete the process within 5 seconds under normal load conditions. 2. System shall limit external calls to the MOE API to a maximum of 5 times per day.
Assumptions:	NA
Notes and Issues:	NA

D. UI Mockups

D.1 Main Query Page



D.2 Selecting Filter Page



D.3 School Details Page

School Profile

School Picture

XXX School

General Info
Location: *69 Serangoon Drive, S423124*
Website: *-> xxxschool.moe.edu.sg*

Details
Type: *Government School*
Nature: *Co-Ed*
SAP / Autonomous / IP: *Autonomous*
Mother Tongue: *Chinese, Malay, Tamil*

Programmes
CCAs:
- **Sports:** *Track & Field, Basketball, Football, Softball, Netball, Hockey, Swimming, ...see more*
- **Performing Arts:** *Guzheng, String Orchestra*
- **Clubs & Societies:** *Weiqi, Mindsports, Chinese Chess*
- **Uniformed Groups:** *Scouts, Girl Guides*

Special Programmes:
- *Some Special Programme*

D.4 Comments Page

XXX School (Comments)

Leave a comment...

I think XXX School is the best school in the west!
Posted 20min ago

No I think YYY School is! They have stronger academic clubs, and they win NSG every year :)
Posted 18min ago

Every school is a good school! Just choose the school nearer to your house :D
Posted 15min ago

→ Reply in thread

Does anybody know how good the performing arts CCAs are?
Posted 28min ago

They're pretty good! String orchestra got Distinction last year
Posted 12min ago

→ Reply in thread

D.5 Compare School Page

Compare Schools

	XXX School	YYY School
Type	Government School	Independent School
Nature	Co-Ed School	Boys School
Travel Time **based on your home location*	20min	1h
Mother Tongue	Chinese, Malay, Tamil	Chinese
Programmes	**Sports:** *Track & Field, Basketball, Football, Softball, Netball, Hockey, Swimming, ...see more* **Performing Arts:** *Guzheng, String Orchestra* **Clubs & Societies:** *Weiqi, Mindsports, Chinese Chess* **Uniformed Groups:** *Scouts, Girl Guides*	**Sports:** *Another Set of Sports* **Performing Arts:** *Guzheng, String Orchestra* **Clubs & Societies:** *Weiqi, Mindsports, Chinese Chess* **Uniformed Groups:** *Scouts, Girl Guides*