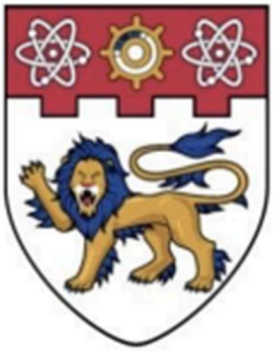

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

SC2006-Software Engineering

Supporting Documents: Functional & Non-Functional Requirements

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 the user firsts loads the home page, or no 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 based on their location.
        1.4.1. The location input shall be a multi-choice dropdown with all locations available.
        1.4.2. The location input shall be empty (no selected locations) by default.
        1.4.3. If 0 locations are provided, the query shall not filter by location.
        1.4.4. If 1 or more locations are provided, the system shall include only schools whose location is in the selected locations in the results.
    1.5. The user shall be able to filter schools based on their Co-Curricular Activities (CCA) offered.
        1.5.1. The CCA input shall be a multi-choice dropdown with all CCA available.
        1.5.2. The CCA input shall be empty (no selected CCA) by default.
        1.5.3. If 0 CCA are provided, the query shall not filter by CCA.
        1.5.4. 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.
    1.6. The user shall be able to filter schools based on their subjects offered.
        1.6.1. The subjects input shall be a multi-choice dropdown with all subjects available.
        1.6.2. The subjects input shall be empty (no selected subjects) by default.
        1.6.3. If 0 subjects are provided, the query shall not filter by subjects.
        1.6.4. 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.
    1.7. The user shall be able to filter schools by cut-off point range.

1.7.1. The cut-off point range shall consist minimum cut-off point and maximum cut-off point.

1.7.2. The minimum cut-off point input shall be an integer within 4-32.

1.7.3. The minimum cut-off point input shall be 4 by default.

1.7.4. The maximum cut-off point input shall be an integer within 4-32.

1.7.5. The maximum cut-off point input shall be larger than or equal to the minimum cut-off point.

1.7.6. The maximum cut-off point input shall be 32 by default.

1.7.7. The system shall include only schools whose cut-off points are within minimum and maximum cut-off point (inclusive) 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 multi-choice dropdown with all levels available.

1.8.2. The level input shall be empty (no selected level) by default.

1.8.3. If 0 levels are provided, the query shall not filter by level.

1.8.4. If 1 or more levels are provided, the system shall include only schools whose level is in the selected levels 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 multi-choice dropdown with all nature codes available.

1.9.2. The nature code input shall be empty (no selected nature code) by default.

1.9.3. If 0 nature codes are provided, the query shall not filter by nature code.

1.9.4. If 1 or more nature codes are provided, the system shall include only schools whose nature code is in the selected nature codes 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 multi-choice dropdown with all types available.

1.10.2. The type input shall be empty (no selected type) by default.

1.10.3. If 0 types are provided, the query shall not filter by type.

1.10.4. If 1 or more types are provided, the system shall include only schools whose type is in the selected types in the results.

1.11. The user shall be able to filter schools based on their session codes

1.11.1. The session code input shall be a multi-choice dropdown with all session codes available.

1.11.2. The session code input shall be empty (no selected session code) by default.

1.11.3. If 0 session codes are provided, the query shall not filter by session code.

1.11.4. If 1 or more session codes are provided, the system shall include only schools whose session code is in the selected session codes 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.
      2.1.7. The system shall display the school's mother tongue.
      2.1.8. The system shall display the school's tags (SAP, Autonomous, Gifted, IP)
   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.
      2.2.7. If the user is authenticated and the user profile includes valid postal code, the system shall display route information from the school to the user's postal code.
         2.2.7.1. The system shall display the computed travel route on the interactive map.
         2.2.7.2. The system shall display the estimated travel time for the selected route.
         2.2.7.3. The system shall provide route information for driving.
         2.2.7.4. The system shall provide route information for cycling.
         2.2.7.5. The system shall provide route information for public transport.
         2.2.7.6. The system shall provide route information for walking.
   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.

2.5. The system shall display the CCAs offered by school.

2.5.1. The system shall display CCAs in ascending lexicographical order.

2.6. The system shall display the MOE programmes offered by school.

2.6.1. The system shall display MOE programmes in ascending lexicographical order.

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.

3.2. The system shall display the attributes of the selected schools side-by-side.

3.3. The user shall be able to initiate school comparison via natural language queries (NLP for comparing schools).

3.3.1. The system shall accept natural-language comparison requests of up to 256 characters.

3.3.2. The system shall process the NLP query to identify target.

3.3.2.1. The system shall display a paragraph regarding the reasons.

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.

5.1.6. If the user is authenticated, the system shall display the user's vote.

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.

7. The user shall be able to edit their user profile.
   7.1. The system shall enforce authentication to edit their user profile.
   7.2. The user shall be able to edit their postal code.
      7.2.1. The postal code shall be a string with 6 digits
      7.2.2. The postal code shall exist in Singapore.

8. The admin shall be able to edit all user's roles.
   8.1. The system shall enforce authentication to edit user roles.
   8.2. The admin shall be able to edit the user role to admin.
   8.3. The admin shall be able to edit the user role to user.
      8.3.1. The user shall not be the last admin.

9. The admin shall be able to edit all schools' cut off points.
   9.1. The system shall enforce authentication to edit school cut-off points.
   9.2. The admin shall be able to edit the school's minimum cut-off point.
      9.2.1. The minimum cut-off point input shall be an integer within 4-32.
   9.3. The admin shall be able to edit the school's maximum cut-off point.
      9.3.1. The maximum cut-off point input shall be an integer within 4-32.
      9.3.2. The maximum cut-off point input shall be larger than or equal to the minimum cut-off point.

A.2 Non-Functional

Usability	1. System shall be intuitive to users with minimal technical knowledge
	    1. 80% of users shall be able to make a simple school search within 2 minutes of using the website
	        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)
	    2. System UI shall adhere to Nielsen Norman group's usability heuristics, with minimalistic design, clear labels and consistent layout
	2. System shall provide accessibility support in compliance with WCAG 2.1 AA guidelines
	    1. System shall focus on screen reader compatibility and keyboard navigation support
	3. System shall provide helpful validation messages and error hints on erroneous user inputs
	4. 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. 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. 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.
	    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:
	    1. What data is collected
	    2. Why it is collected
	    3. How long it will be retained
	    4. How it will be used
	5. System shall only collect personal data with the user's explicit consent.
	    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. Actions that are considered violations.
	    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. Unit test coverage should be >80% 2. System shall undergo user acceptance testing (UAT) before release with >10 representative users 1. UAT shall pass with >90% task completion and >80% user satisfaction rating