
Use Cases

for

<SchoolFinder>

Version 1.0

Prepared by: Cheng Han, Abhiyan Singh, Ying Sheng,
Jonathan Yap, Jia le, Boon Khang

<SchoolFinder>

<01-02-21>

Revision History

Name	Date	Reason For Changes	Version
Cheng Han	01-02-21	Adding more details	1.0

Use Case Template

Use Case ID:	UC001		
Use Case Name:	Login		
Created By:	Cheng Han	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, User database
Description:	Authenticate the user
Preconditions:	The user has downloaded and started the application
Postconditions:	1) User is able to login. 2) User is unable to login.
Priority:	High
Frequency of Use:	High
Flow of Events:	1) User inputs 'EMAIL' and 'PASSWORD'. 2) User clicks the login button. 3) System interacts with the user database and authenticates the user. 4) User is logged in.
Alternative Flows:	AF1-S3: If the password is wrong 1) User is prompted to re-enter the password. AF2-S3: If the email is wrong 1) User is shown 'EMAIL DOES NOT EXIST'.
Exceptions:	EX1: Local server storing the user database is down
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC002		
Use Case Name:	Register		
Created By:	Boon Khang	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, User Database
Description:	Register user into user database.
Preconditions:	User does not have an account.
Postconditions:	User account is created.
Priority:	Medium
Frequency of Use:	Medium

Flow of Events:	<ol style="list-style-type: none"> 1) User selects the 'REGISTER' button. 2) User is prompted to enter 'EMAIL', 'PASSWORD' and 'CONFIRMATION OF PASSWORD'. 3) System checks whether the user is already registered. 4) System creates a new user. 5) User is prompted to login.
Alternative Flows:	<p>AF-S2: If user inputs invalid password format</p> <ol style="list-style-type: none"> 1) System prompts the user to enter password again. <p>AF-S3: If user already exist</p> <ol style="list-style-type: none"> 1) System informs the user 'USER ALREADY EXISTS' 2) User is prompted to enter another email address.
Exceptions:	EX1: Local server storing the user database is down.
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC003		
Use Case Name:	Favourite		
Created By:	Abhiyan Singh	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Favourites Database
Description:	Add or remove a school from the Favourite Section.
Preconditions:	User is logged in.
Postconditions:	School is added in the Favourite Database.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1) If the heart icon beside the name of the school is non-coloured, user clicks on the heart icon. 2) System stores the name of the school and username in the Favourites database. 3) The empty heart icon becomes pink in colour.
Alternative Flows:	<p>AF1-S1: If the heart icon is already pink</p> <ol style="list-style-type: none"> 1) System removes the school from the Favourites database. 2) Pink heart icon becomes empty.
Exceptions:	EX1: Local server storing the user database is down.
Includes:	-
Special Requirements:	-
Assumptions:	User is already logged in.
Notes and Issues:	-

Use Case ID:	UC004		
Use Case Name:	Report		
Created By:	Jonathan Yap	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Report Database		
Description:	User submits a report on abusive behaviour by other users		
Preconditions:	User is logged in		
Postconditions:	Report on abusive user is added to the report database		
Priority:	Low		
Frequency of Use:	Low		
Flow of Events:	<ol style="list-style-type: none"> 1) The user clicks on the report button on another user's review. 2) The user then types out a brief description on what the offending user did. 3) System stores the report into the database. 		
Alternative Flows:	-		
Exceptions:	1) If User has already rated the school, rating it again will remove the old rating given by the user.		
Includes:	EX1: Local server storing the user database is down.		
Special Requirements:	-		
Assumptions:	All Servers are up and running with no connection problems.		
Notes and Issues:	-		

Use Case ID:	UC005		
Use Case Name:	Review		
Created By:	Jia Le	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Government Database, Comments Database		
Description:	Post comment and rating for the school		
Preconditions:	<ol style="list-style-type: none"> 1) User is logged in 2) User must be at the 'MORE INFORMATION' page. 		
Postconditions:	User comment and rating successfully posted.		
Priority:	Medium		
Frequency of Use:	Low		
Flow of Events:	<ol style="list-style-type: none"> 1) User inputs personal comment. 2) User inputs rating. 3) User selects 'SUBMIT REVIEW'. 4) System gets the school name from the government database. 5) System stores comment and rating to the comments database with user name and school name as the key. 6) User is prompted that the comment and rating has been successfully posted. 		

	7) Use case ends.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC006		
Use Case Name:	Search		
Created By:	Ying Sheng	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Government API (Data.gov.sg), User database, Google API, comments API
Description:	Basic search for a specific school
Preconditions:	User is at the homepage
Postconditions:	Display the search results
Priority:	High
Frequency of Use:	High
Flow of Events	<ol style="list-style-type: none"> 1) User types in the name of the school to search for. 2) System interacts with the government API to retrieve a list of schools based on the input. 3) System displays a list of schools to the user. 4) User selects the 'MORE INFORMATION' button. 5) System retrieves school information from the government API and school's reviews from the review database. 6) System displays the detailed school page. 7) User can read the reviews by the other users.
Alternative Flows:	<p>AF1-S2: If no matching school is found.</p> <ol style="list-style-type: none"> 1) System shows a "NO SCHOOL IS FOUND" message. 2) Return to step 1. <p>AF2-S3: If the user is logged in.</p> <ol style="list-style-type: none"> 1) System retrieves the user's postal code from the user database. 2) System retrieves the school's postal code from the government API. 3) System will interact with Google API to find the distance between the school and the user's postal code. 4) System will display the distance of the schools from the user's postal code along with the school's name in the list 5) Return to step 4. <p>AF3-S6: If the user is logged in.</p>

	1) User can add the school as favourite (UC003) 2) User can post a review for the school (UC005). 3) User can report inappropriate comments (UC004).
Exceptions:	EX1: Local server storing the user database is down.
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC007		
Use Case Name:	Advanced Search		
Created By:	Cheng Han	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Government Database (Data.gov.sg), User database, Google API, comment database.
Description:	Advanced search with filters on cut off points, CCAs, special programs.
Preconditions:	User is at the homepage
Postconditions:	Displays a list of the schools which match the filters
Priority:	High
Frequency of Use:	High
Flow of Events:	1) User clicks on the 'ADVANCE SEARCH' button. 2) User selects the filters they desire and press the 'APPLY' button. 3) System interacts with the government API to retrieve a list of schools based on the input. 4) System displays a list of schools to the user. 5) User selects the 'MORE INFORMATION' button. 6) System retrieves school information from the government API and school's reviews from the review database. 7) System displays the detailed school page. 8) User can read the reviews by the other users.
Alternative Flows:	AF1-S3: If no matching school is found. 3) System shows a "NO SCHOOL IS FOUND" message.

	<p>4) Return to step 1.</p> <p>AF2-S4: If the user is logged in.</p> <p>6) System retrieves the user's postal code from the user database.</p> <p>7) System retrieves the school's postal code from the government API.</p> <p>8) System will interact with Google API to find the distance between the school and the user's postal code.</p> <p>9) System will display the distance of the schools from the user's postal code along with the school's name in the list</p> <p>10) Return to step 4.</p> <p>AF3-S7: If the user is logged in.</p> <p>4) User can add the school as favourite (UC003)</p> <p>5) User can post a review for the school (UC005).</p> <p>6) User can report inappropriate comments (UC004).</p>
Exceptions:	EX1: Local server storing the user database is down.
Includes:	
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-

Use Case ID:	UC008		
Use Case Name:	More Information		
Created By:	Boon Khang	Last Updated By:	-
Date Created:	01-02-21	Date Last Updated:	-

Actor:	User, Government Database (Data.gov.sg), Google API, Comment Database
Description:	More information on the specified school
Preconditions:	System has generated a list of schools
Postconditions:	Displays a detailed school page
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<p>1) User selects a school from the list by clicking on the 'MORE INFORMATION' button.</p> <p>2) System retrieves school information from the government API and school's reviews from the review database.</p> <p>3) System displays the detailed school page.</p>
Alternative Flows:	-
Exceptions:	EX1: Local server storing the user database is down.
Includes:	-
Special Requirements:	-
Assumptions:	All Servers are up and running with no connection problems.
Notes and Issues:	-