Use Cases

for

<Project>

Version 1.0 approved

Prepared by Sebastian & Alicia

Team of 5

29 January 2022

Revision History

Name	Date	Reason For Changes	Version

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

1. Use Case Identification

1.1. Use Case ID

Give each use case a unique numeric identifier, in hierarchical form: X.Y. Related use cases can be grouped in the hierarchy. Functional requirements can be traced back to a labeled use case.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.3.1 Created By

Supply the name of the person who initially documented this use case.

1.3.2 Date Created

Enter the date on which the use case was initially documented.

1.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

2. Use Case Definition

2.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor(s) that will be performing this use case.

2.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

2.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

2.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

2.5. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

2.6. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

2.7. Flow of Events

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system.

2.8. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative course, and describe any differences in the sequence of steps that take place. Number each alternative course using the Use Case ID as a prefix, followed by "AC" to indicate "Alternative Course". Example: X.Y.AC.1.

2.9. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use

case execution fails for some unanticipated reason. Number each exception using the Use Case ID as a prefix, followed by "EX" to indicate "Exception". Example: X.Y.EX.1.

2.10. Includes

List any other use cases that are included ("called") by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

2.11. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

2.12. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

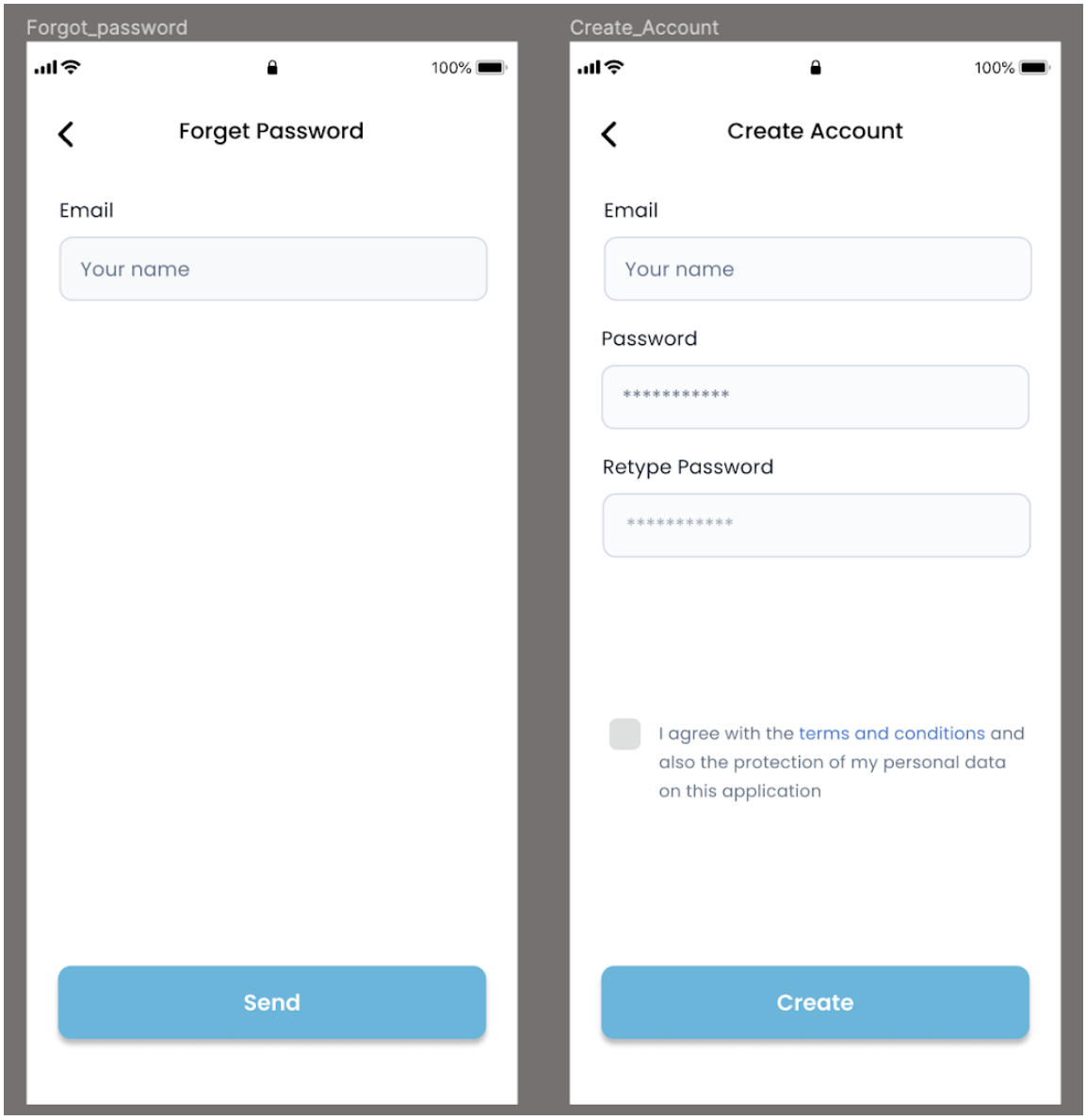
2.13. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case Template

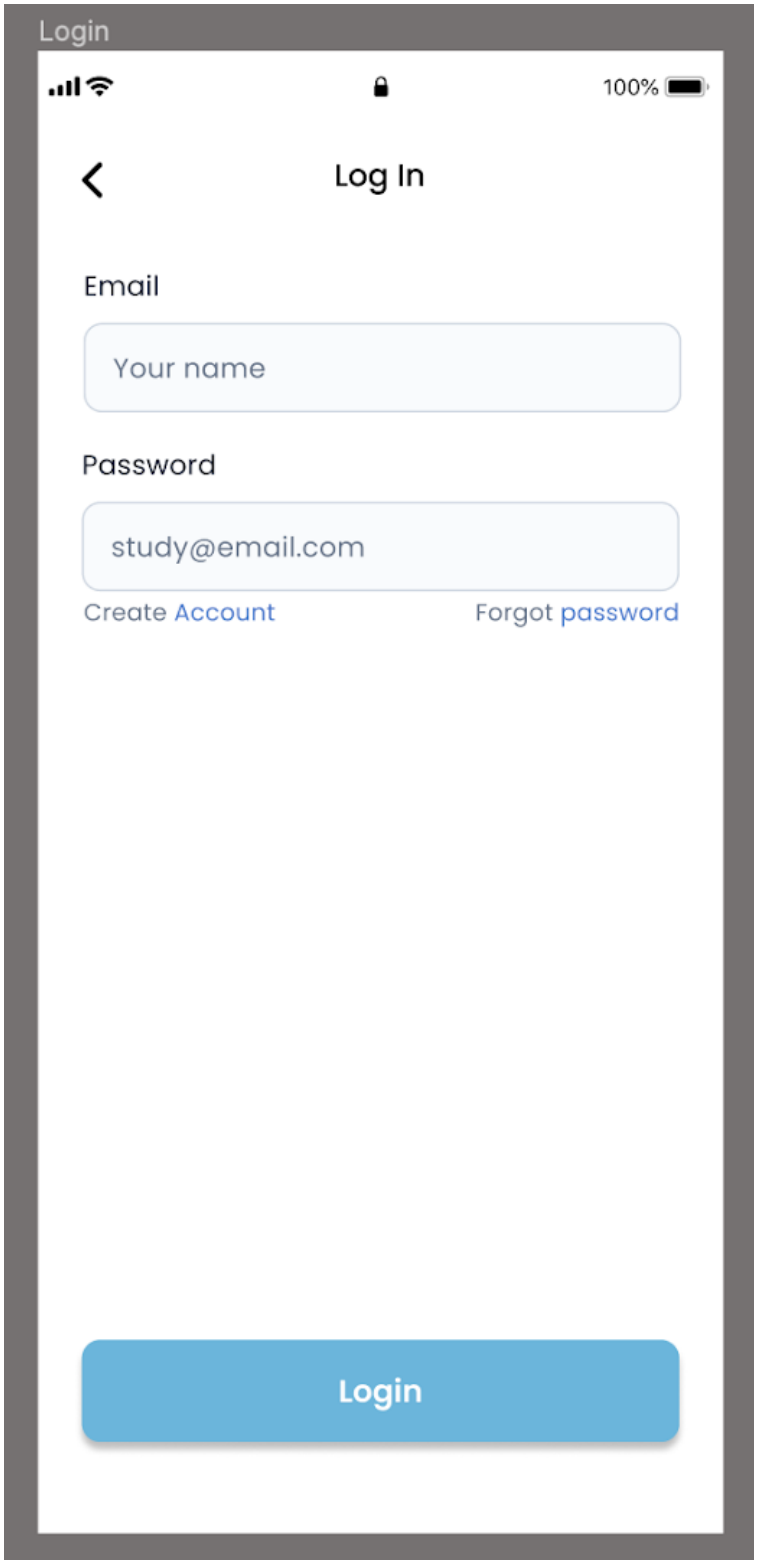
Use Case ID:	001		
Use Case Name:	Create Account		
Created By:	Sebastian, Alicia	Last Updated By:	Alicia
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Account System
Description:	The user is creating an account
Preconditions:	1. The User must not have an account.
Postconditions:	1. The User has created an account.
Priority:	
Frequency of Use:	
Flow of Events:	1. The User selects “Create Account” button 2. The application requests for the User’s username, email and password 3. The User enters their username, email and password 4. The User ticks the terms and conditions box. 5. The User selects the “Register” button 6. The application checks if there are any existing users with the same email and if not, stores the information with the Account System 7. The application then confirms with the user that their account has been created and prompts the user to login.
Alternative Flows:	AF1-S5: The User has not ticked the terms and conditions box. 1. The application would display “Unchecked Terms and Conditions.” 2. The application returns to step 2. AF2-S6: There is an existing user with the same email 1. The application would display “Email already in use.” 2. The application returns to step 2. AF3-S6: The email format is wrong. 1. The application would display “Invalid Email Format.” 2. The application returns to step 2. AF4-S7: The user has not verified his email. 1. The application would display “Email not verified.” 2. The application returns to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	1. The User has installed the application.
Notes and Issues:	-


Forgot_password
100%
Forget Password
Email
Your name
Send
Create_Account
100%
Create Account
Email
Your name
Password
****
Retype Password
****
I agree with the terms and conditions and also the protection of my personal data on this application
Create

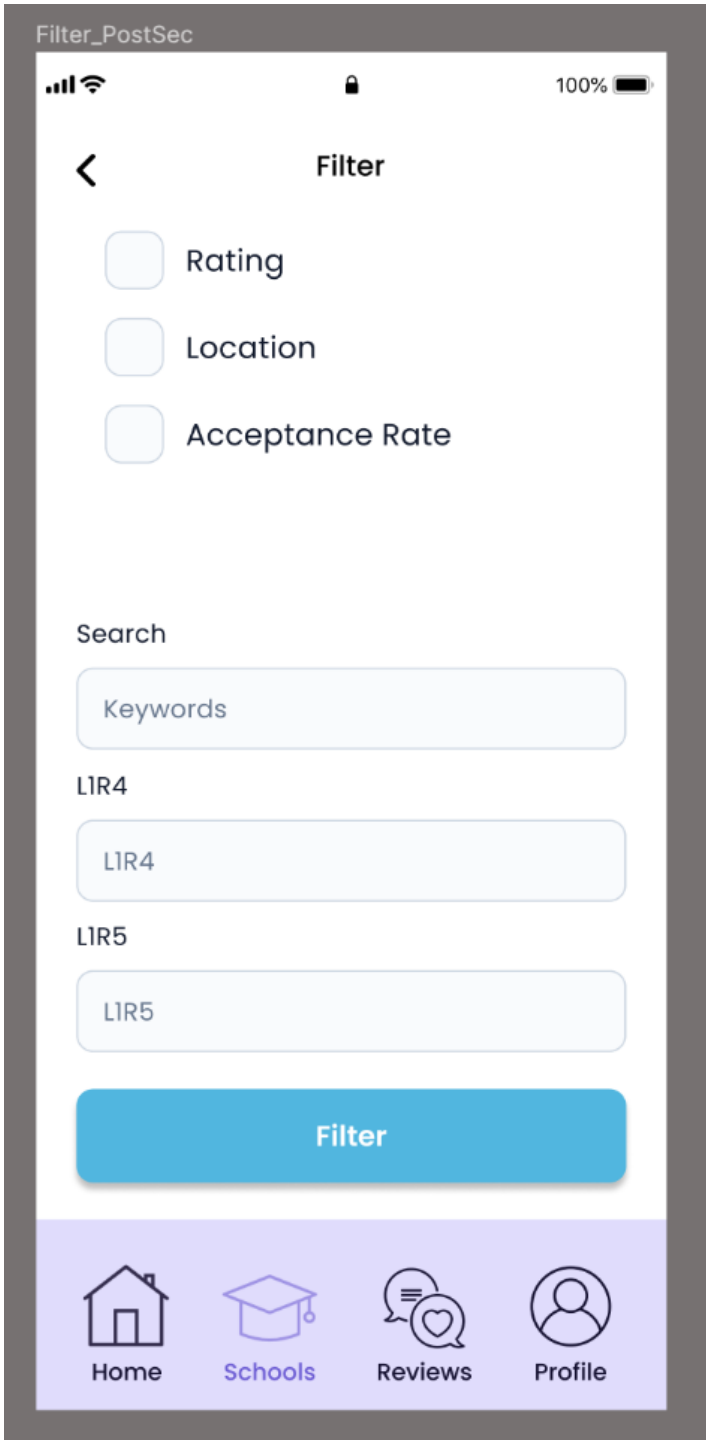
Use Case ID:	002		
Use Case Name:	Login		
Created By:	Sebastian, Alicia	Last Updated By:	Alicia
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Account System
Description:	The user is logging into the application
Preconditions:	1. The User has an account with the Account System. 2. The User must not be logged in.
Postconditions:	1. The User is logged in.
Priority:	
Frequency of Use:	
Flow of Events:	1. The application requests the User's login information 2. The User enters their Email and Password 3. The application verifies the information with the Account System 4. If the information is valid the user is logged in
Alternative Flows:	AF1-S4: The Email does not exist. 1. The application would display "Account does not exist." 2. The application returns to step 2. AF2-S4: The Password is incorrect. 1. The application would display "Wrong Password." 2. The application returns to step 2. AF3-S4: The password is inputted incorrectly 3 times. 1. The application would display "Reset password?" 2. The application returns to step 2. AF4-S4: The user did not input the email or the password when logging in. 1. The application would display "Missing Email" or "Missing Password" respectively. 2. The application returns to step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-


Login
100%
Log In
Email
Your name
Password
study@email.com
Create Account
Forgot password
Login

Use Case ID:	003		
Use Case Name:	Filter by Location		
Created By:	Sebastian, Alicia	Last Updated By:	Alicia
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Cloud
Description:	The user is filtering the suggestions based on their location
Preconditions:	1. The User has an account with the Account System 2. The User is logged in
Postconditions:	1. The suggestions given is based on the User's location
Priority:	-
Frequency of Use:	-
Flow of Events:	1. The User selects the option to filter by location 2. The application would request the User to input their location 3. The User inputs their location 4. The application then filters the suggested schools based on the User's location.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

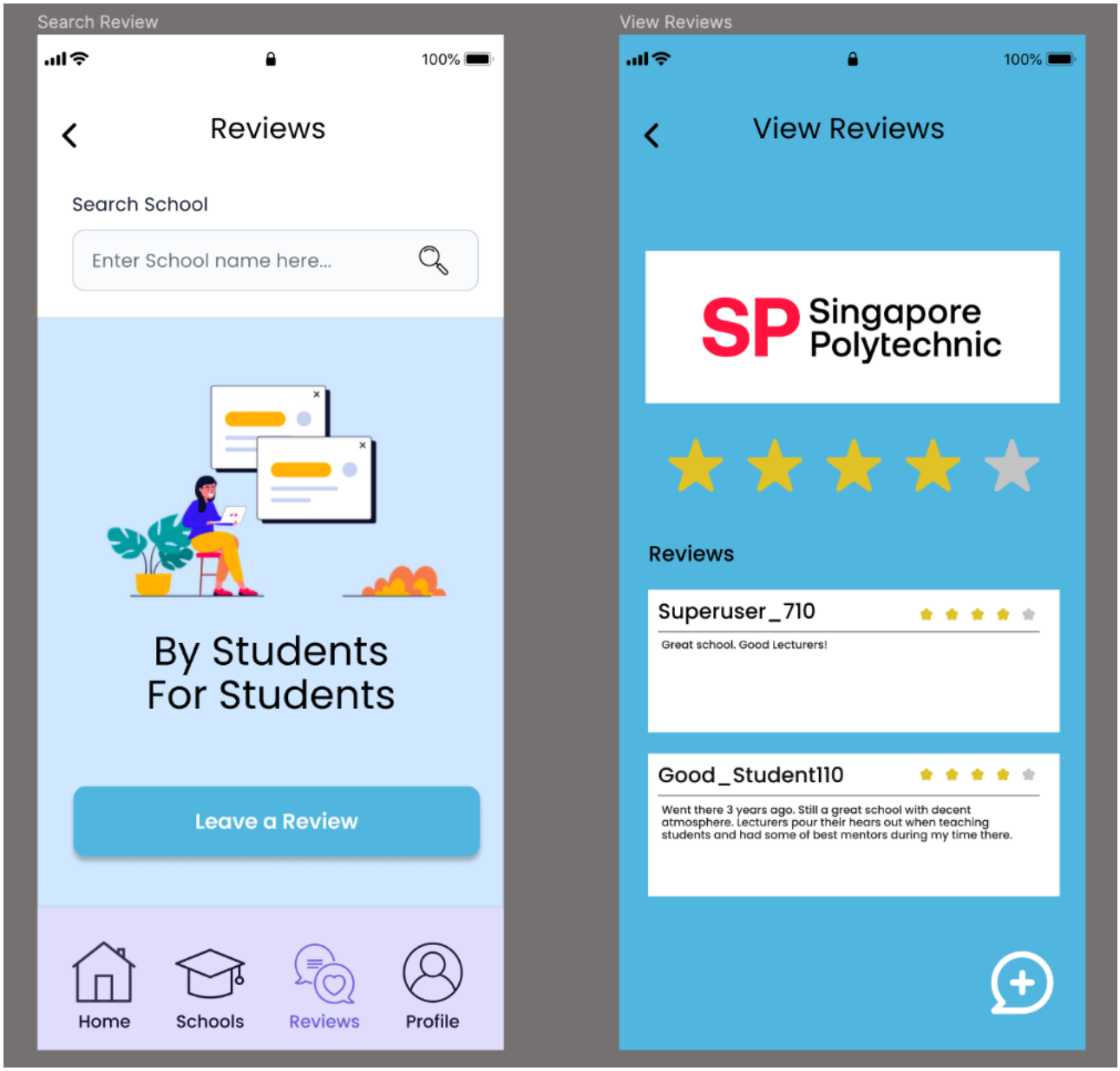

Filter_PostSec
100%
Filter
Rating
Location
Acceptance Rate
Search
Keywords
L1R4
L1R4
L1R5
L1R5
Filter
Home
Schools
Reviews
Profile

Use Case ID:	004		
Use Case Name:	Filter by Results		
Created By:	Sebastian, Alicia	Last Updated By:	Alicia
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Cloud
Description:	The user is filtering the suggestions based on their academic results
Preconditions:	1. The User has an account with the Account System 2. The User is logged in
Postconditions:	1. The suggestions are filtered based on the User's results
Priority:	
Frequency of Use:	
Flow of Events:	1. The User selects the option to filter by results 2. The application would request the User to input their results 3. The User inputs their results 4. The application then filters the suggested schools based on the User's results.
Alternative Flows:	AF-S3: The results entered are invalid (not within the specified range). 1. The application would not display any schools.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	Invalid results refer to results that are out of range. For example, a GPA higher than 5.

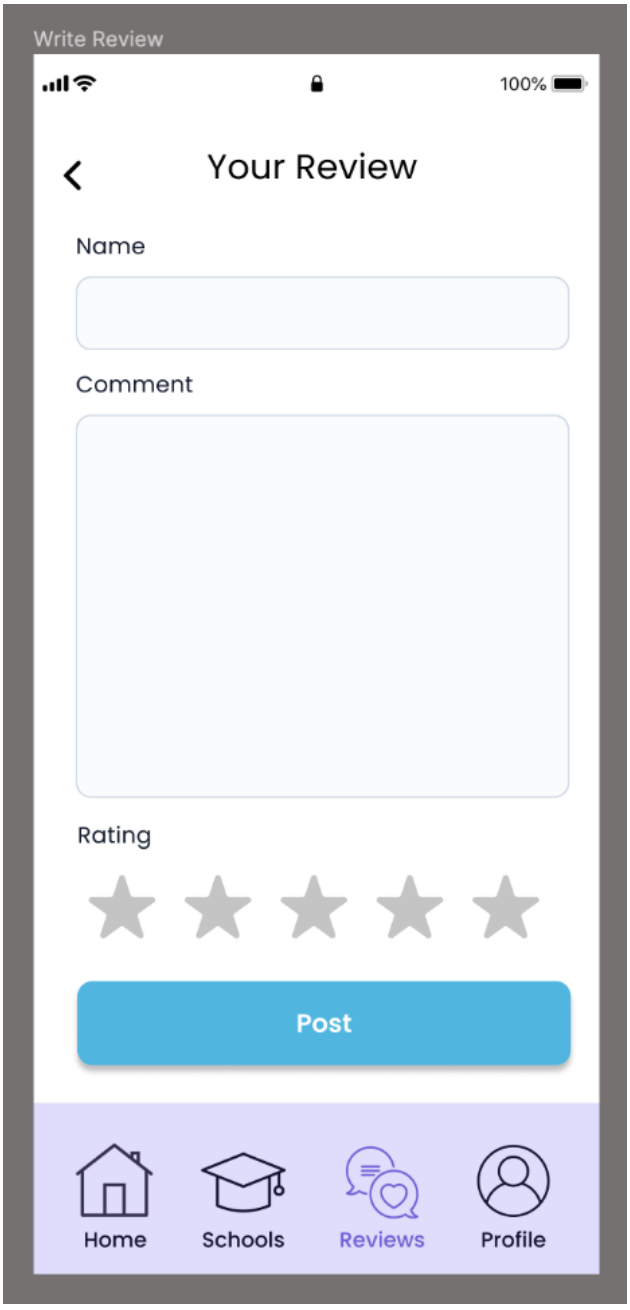
Use Case ID:	005		
Use Case Name:	View Review		
Created By:	Sebastian	Last Updated By:	Sebastian
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Cloud, Account System
Description:	The user is viewing a review on a course or school.
Preconditions:	1. The User has an account with the Account System 2. The User is logged in
Postconditions:	1. Reviews on a course/school's page are seen by the User
Priority:	
Frequency of Use:	
Flow of Events:	1. The User enters the School for which they want to view the review. 2. The system verifies that the School exists. 3. The application would show all the reviews left by users on the course/school.
Alternative Flows:	AF-S2: School does not exist 1. The application would not display any schools.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	


Search Review
100%
Reviews
Search School
Enter School name here...
By Students
For Students
Leave a Review
Home
Schools
Reviews
Profile
View Reviews
100%
View Reviews
SP Singapore Polytechnic
Reviews
Superuser_710
Great school. Good Lecturers!
Good_Student110
Went there 3 years ago. Still a great school with decent atmosphere. Lecturers pour their hears out when teaching students and had some of best mentors during my time there.

Use Case ID:	006		
Use Case Name:	Give Review		
Created By:	Sebastian	Last Updated By:	Sebastian
Date Created:	29/1/22	Date Last Updated:	29/1/22

Actor:	User (Initiating Actor), Cloud, Account System
Description:	The user is leaving a review on a course or school.
Preconditions:	1. The User has an account with the Account System
Postconditions:	1. A review by the User is left on the course/school page on the application
Priority:	
Frequency of Use:	
Flow of Events:	1. The User selects the School to give a review. 2. The system verifies that the School exists. 3. The application would request the User to type out their review. 4. The User types out their review and submits it. 5. The review is then sent and stored in the cloud.
Alternative Flows:	AF-S2: School does not exist 1. The application will not display any schools.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	


Write Review
100%
Your Review
Name
Comment
Rating
Post
Home
Schools
Reviews
Profile