

CZ2006 Project

Summary

Our goal is to develop an app that makes university applications for Singaporeans much easier. The app should allow users ease of access to information about universities, through search and filter functions that allows users to find universities that fit their requirements. Secondly, our app will also ask for user's academics and conditions, and attempt to match a suitable and eligible university for them based on whether they have a good chance of getting an offer in the university.

Functional Requirements

Registration:

User must be allowed to create an account with their email address

User must create a 6-18 character length password to register their account.

System must validate the email address and password.

User must be allowed to retrieve their password by clicking on "Forgot Password".

System must prompt the user to enter their email address to send the user a link to change password.

System must send an email with a verification code to the email address of the user to allow them to change password.

User must be allowed to input the following information during account registration:

- Name
- Date of Birth
- Nationality
- Educational background
- Address

Login Page:

The user must be able to login with their email address and password

System must confirm that login information is correct

System must display an error message to the user to re-enter their email address and password if either of it is confirmed to be incorrect

System must log the user in when login details are validated as correct.

System must redirect the user to the home page when login is successful.

Navigation:

The user must be able to search for schools using the following filters:

- Cut-off points
- Ranking of the school
- Courses offered by the school
- Distance of school from where they live (Location of school)

The system must sort the search results according to the user inputs in the filters.

User must be allowed to retrieve the information of the school selected. This information includes:

- Introduction to school
- Courses offered by school
- Reviews of school by former students
- Location of school (Estate/Neighbourhood)

User must be allowed to retrieve information of the course selected under each school. This information includes:

- Admission requirements (Cut off points)
- Curriculum
- Course duration and structure
- Career prospects

User must be allowed to download a PDF brochure containing the course information on the app.

Any distance measurement under Location must be formatted as a number with an accuracy of 2 decimal places and the measurement unit "km"

Main Page:

Users must be able to search for universities based on the courses they are looking for.

Users must be able to search for universities based on the country the university is in.

The system must recommend to the user the universities and schools that fit other users criteria the most number of times.

The system must recommend to the user the most-seeked courses by other users using the search function

The system must display the latest news happening in universities in a news feed section.(to be cfm)

Reviews and Ratings:

Users should be able to leave comments under a university, specifying either the faculty or under "general comment".

Users should be able to rate their current university 1-5 stars out of 5

Users should be able to view all comments and reviews left by other users of a university, and be able to filter out comments based on faculty/category.

Non-functional Requirements

Performance

System should not crash when the application is open or running.

System should display a user-selected page or information within 5 seconds of clicking.

System should only update overnight (SGT) and finish updating before 8am.

System should be able to maintain and handle high workload

Security

System must ensure the integrity of the user account information

System must behave in a predictable and correct manner

System should mask the password field

Usability

System must allow ease of returning to the previous page being browsed by the user.

System must display error messages and feedback when the user makes an incorrect input.

System must have a consistent layout (text size, colour, font)

Compatibility

System must be compatible mainstream browsers such as Chrome and Mozilla Firefox

Reliability

System must be able to restore to full functionality within 5 minutes of system reboot.

UI Mock-ups

Your Future

Full Name:

Email:

Password:

Confirm Pw:

[Register](#)

Your Future

Full Name:

Email:

Password:

[Login](#)

Figure 1. Login and Registration page

Your Future.com

Sign up Log In

Your Future.com


★ Country ▾

🏠 Ranking ▾

📖 Course ▾


[Let's Go](#)

Most Searched Universities by other users



Cambridge University
★★★★★
England • \$\$\$\$ • Cambridge


Most Searched Universities



Cambridge University
★★★★★
England • \$\$\$\$ • Cambridge
📊 QS World Ranking: 3

[See Details](#)

Up and Coming Universitie



The University of Tokyo

Figure 2. Main Page



Your Future.com

Sign up

Log In

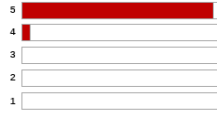


Cambridge University

Reviews by alumnus and students

Overall ratings and reviews

★★★★★ 4.8/5
based on 642 ratings



Leave a review

☆☆☆☆☆

Review

Filter by faculty/program



Steven

★★★★★ • Reviewed 3 days ago

The University of Cambridge offers a very rigorous program between lectures and supervisions. The terms are 8 weeks (starting on a Thursday) and they pack a lot of information...

[Read More +](#)



★★★★★ • Reviewed 3 days ago

Program between lectures and supervisions. The

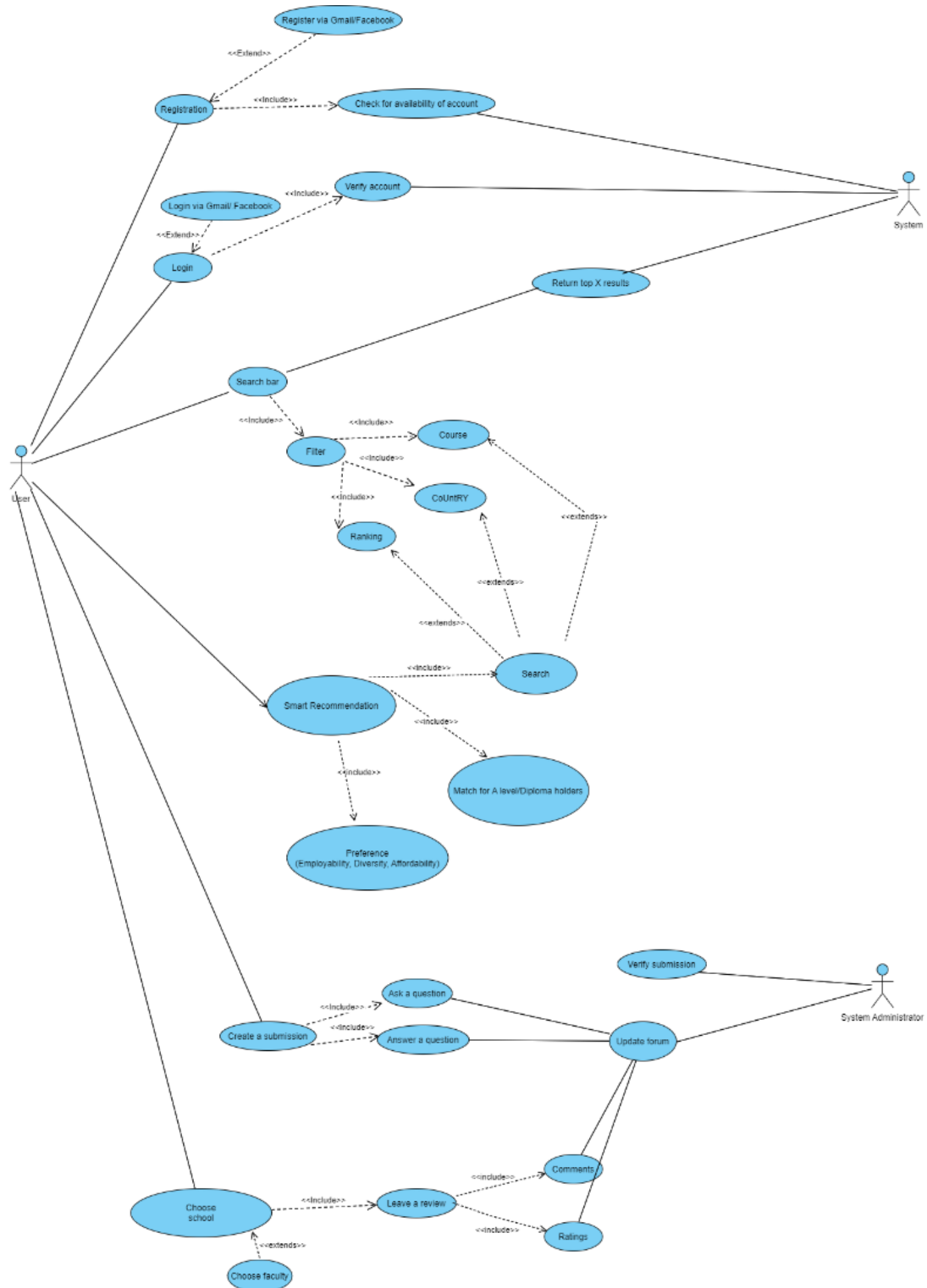
Figure 3. Review Page

Data Dictionary

Name	Description	Data type
Administrator	<ul style="list-style-type: none"> An administrator is a person who manages the application Administrators can log on to administrator mode 	object
userList	<ul style="list-style-type: none"> Contains registered users and their information 	array
User	<ul style="list-style-type: none"> A user is a person who signed up for the application Users use the application in the user mode Each user must have its username, password and email address Users must choose to be one of Freshmen/Undergraduate/Postgraduate/Teacher when signing up Users can register his/her education status and address when signing up Users can click 'like' button to put his/her schools of interest in a list to check any changes 	object
username	<ul style="list-style-type: none"> A user must have a unique username which consists of uppercase / lowercase characters or integers Username must be at least 5 characters long Required attribute 	String
password	<ul style="list-style-type: none"> Users must have at least one uppercase / lowercase / special characters / integers in their passwords Password must be at least 6 characters long Can be reset / changed using resetPassword / changePassword functions Required attribute 	String
email	<ul style="list-style-type: none"> Email address that the user uses to sign in Should be in the valid email address format Required attribute 	String
state	<ul style="list-style-type: none"> Stores the state of the user (Freshmen/Undergraduate/Postgraduate/Teacher) Optional attribute 	enumeration
address	<ul style="list-style-type: none"> Must be a valid, existing address on Google Map 	String
likedSchool	<ul style="list-style-type: none"> Stores the schools which the user liked 	array
schoolList	<ul style="list-style-type: none"> Contains all the schools in the system 	array
School	<ul style="list-style-type: none"> A school has rating / reviews / ranking / cut-off points / courses it offers / its location as attributes 	object
Signup	<ul style="list-style-type: none"> Consists of textboxes for username, email address, password (required), home address, education status (optional) 	object
Login	<ul style="list-style-type: none"> Consists of two textboxes for email address and password, 'log in' and 'forgot password?' buttons Has a function that checks if login information is in the userList → if exists, change isLoggedIn status to true and redirect to home page / else, give a pop-up page with an error message 	object
isLoggedIn	<ul style="list-style-type: none"> Login status 	boolean

	<ul style="list-style-type: none"> • If isLoggedIn == true, the application is in user mode • Else, shows guest mode 	
schoolList	<ul style="list-style-type: none"> • Contains all the schools in the system 	array
School	<ul style="list-style-type: none"> • A school has recommended / ranking / cutoff points / courses it offers / its location as attributes 	object
rating	<ul style="list-style-type: none"> • When a user clicks the 'recommend' button of a school in the search page, integer attribute 'recommended' increments by 1 	integer
ranking	<ul style="list-style-type: none"> • The school's ranking according to the QS University Ranking, either generic or by subject depending on the user's choice 	integer
program	<ul style="list-style-type: none"> • A program is an area of study offered by a university. 	dictionary
course	<ul style="list-style-type: none"> • Contains all the courses offered by the university, cut-off points and maximum capacity of each course 	dictionary
location	<ul style="list-style-type: none"> • Country / city that the school belongs to 	String
schoolFee	<ul style="list-style-type: none"> • School fee per semester 	float
Google map (& GPS)	<ul style="list-style-type: none"> • Detects users' current location and show the nearest schools around the user's current location • Show schools near the address that the user searches for 	JSON (API)
Filter	<ul style="list-style-type: none"> • Filter is a feature that processes search words to exclude schools not wanted by the user / sort the schools in order of distance, fees or ranking 	object
Search	<ul style="list-style-type: none"> • Search universities in the schoolList according to the matching rate with the search word 	object
Review	<ul style="list-style-type: none"> • Review of a school on the app by a verified student or alumni user, including a comment and a rating of the school 	object

Use Case Diagram



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Leaving Review of a university and/or course		
Created By:	Sharan	Last Updated By:	Sharan
Date Created:	6/2/2022	Date Last Updated:	6/2/2022

Actor:	User, Review system
Description:	User leave a review of a university
Preconditions:	User must be registered and logged in to his/her account
Postconditions:	User must be enrolled in the school to post his review
Flow of Events:	<ol style="list-style-type: none">1. User click on 'leave a review' to post feedback on a university2. User leave a rating [on a scale of 1 to 5] by selecting 1-5 number of stars3. User write a comment4. User choose a category for the review (e.g. "School of Computer Science")5. System check if user's profile is a enrolled student or alumni of the school, if yes, user's review is uploaded
Alternative Flows:	<ol style="list-style-type: none">1. User is not a student or alumni of reviewed school2. System prompts an error message and redirect user to home page

Use Case ID:	2		
Use Case Name:	User finds recommendation of university using the site's recommendation tool		
Created By:	Kay	Last Updated By:	Kay
Date Created:	6/2/2022	Date Last Updated:	6/2/2022

Actor:	User, System
Description:	User goes through a series of survey to let system know of their requirements and details, which the system will use algorithms to give recommendations of universities to the user.
Preconditions:	User must be registered and logged in to his/her account
Postconditions:	User finishes survey and match, and system comes up with recommendation (possible to have no matching universities)
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on "Recommendation Tool" button 2. System Checks if user is logged in, if yes, directs user to a survey 3. User types in search requirements like Course, country, start-year, before clicking on "Next" 4. System check that user filled in all required fields, if yes, redirect user to "personal match" page 5. Users fill in information about themselves like prior study level, test results, grades, before clicking on "Next" 6. System check that user filled in all required fields, if yes, direct to "preference page" 7. Users fill in preferences on a scale of 1 to 10 using a sliding tool for categories including "school reputation", "cost", "convenience", and click on "find recommendation" 8. System matches relevant information user gave and recommends schools to the user from its database.
Alternative Flows:	<ol style="list-style-type: none"> 1. User clicks on "Recommendation Tool" button 2. System Checks if user is logged in, if yes, directs user to a survey 3. User exits survey by clicking on home page icon 4. System saves user's choices for future resuming
Exceptions:	<ol style="list-style-type: none"> 1. System checks if user is logged in, if no, system prompts user to log in before using the function

Use Case ID:	3		
Use Case Name:	User use home page's search function to find schools		
Created By:	Kay	Last Updated By:	Kay
Date Created:	6/2/2022	Date Last Updated:	8/2/2022

Actor:	User, System
Description:	User use the home page's search and filter interface to search for universities that satisfy a few conditions
Preconditions:	User must be registered and logged in to his/her account
Postconditions:	User presses the search button and gets results
Flow of Events:	<ol style="list-style-type: none"> 1. User selects course "computer science" from the "course" drop-down filter 2. User selects "United States" from the "country" filter 3. User selects "Within top 100" from the "ranking" filter 4. User clicks search button 5. System comes up with corresponding school results
Alternative Flows:	<ol style="list-style-type: none"> 1. User types in partial name of university 2. System matches name and finds corresponding universities from database 3. System displays them real time in the home page without user clicking on search button
Exceptions:	- No schools match user's requirement, system prompts a pop-up message to tell user no matches were found.