

Software Requirements Specification for c4me

Prepared by

Team Bingo:

Bingling Dong

Omar Amin

Jafree Khan

Olesia Elfimova

Based on IEEE Std 830-1998 (R2009) document format

Copyright © 2020 Team

1. Introduction

c4me helps students decide where to apply for college. Its functionality is similar to several successful commercial software systems such as Naviance (licensed by many high schools) and commercial websites such as CollegeData.com (a service provided by 1st Financial Bank). c4me is for students planning to get a Bachelor's degree.

1.1 Purpose

The purpose of this document is to specify how **c4me** system should operate. The intended audience for this document is all the members of the development team, from the instructors to the software engineers and designers. This document serves as an agreement among all parties and as a reference for how the college recommendation tool should be constructed.

1.2 Scope

For this project the goal is for students to easily find a college that fits their needs. There will be available only colleges in the U.S., but the site should be usable for students from any state.

1.3 Definitions, acronyms, and abbreviations

c4me	Colleges for Me
IEEE	Institute of Electrical and Electronics Engineers

1.4 References

IEEE Std 830-1998 (R2009) - IEEE Recommended Practice for Software Requirements Specification.

1.5 Organization

This Software Requirements Specification document is divided into multiple sections. The first section includes explanations of the purpose, scope and organization of the document. The first section also handles the description of project-specific words and references. The second section of this document encompasses functionality, use cases, and the use case diagram. The third section is an enumerated listing of all other requirements described for this system. The fourth section provides a table of contents and contributions of team members.

2. Overall Description

2.1 Functionality for Students

2.1.1 Create Account

This allows a new user to create an account by choosing a username and password. This does not require administrator approval.

2.1.2 View/Edit Profile

A student can view and edit all information in his/her profile (except 'userid' cannot be edited), e.g., to update the status of a college application.

[2020-01-31: updated previous sentence] When a student enters an acceptance decision, the system checks for statistical consistency between that decision and other information in the student's profile. If the acceptance decision is statistically unlikely based on other information in the student's profile, the system marks it as questionable and indicates this to the student. Questionable acceptance decisions are ignored by the system when computing all query results; for example, they are omitted from scatterplots. This provides some benefit even though information in student profiles is self-reported, provided most students are honest. If a student submits fake GPA, fake test scores, and fake acceptance decisions that are, taken together, statistically similar to correct data in honest students' profiles), the fake data won't significantly skew query results.

2.1.3 Search for Colleges

This displays a list of colleges satisfying the specified search criteria, a.k.a. filters. The user can specify whether to use a strict or lax interpretation of filter conditions. Strict means that filter conditions involving missing data are treated as false; lax, that they are treated as true. Each filter does nothing in its default state; it limits the search results only if the user sets another value for it. The list can be sorted in various ways, including by name, admission rate, cost of attendance, and ranking. The following information is provided for each college, either directly in the list, or by selecting a college in the list to view additional information about it: institution type (public, private nonprofit, or private for-profit, from College Scorecard field CONTROL), admission rate, completion rate (from CollegeData.com field "Students graduating within 4 years"), cost of attendance, median completed student debt (from College Scorecard field GRAD_DEBT_MDN), ranking, and size. The student can modify and re-run the current search (you are free to display search results on the same or a different page than the search boxes, whichever makes this feature easier to implement). [2020-01-31: added previous sentence.]

Supported filters include:

- admission rate: range for percentage of applicants admitted. The system gets admission rate from College Scorecard field ADM_RATE.
- cost of attendance: upper bound on the cost of attendance. For public state institutions, the in-state or out-of-state cost should be used as appropriate, based on the user's state of residence. The system gets cost of attendance from CollegeData.com.
- location: a list of states or a region in which the college should be located. A region is a shorthand for a set of states. For simplicity, use the four regions defined by the U.S. Census Bureau: Northeast, Midwest, South, and West.
- majors: up to 2 majors that should be offered. The system gets the list of undergraduate majors at a college from CollegeData.com. CollegeData.com sometimes uses long names for majors, e.g., "English Language and Literature, General", and "Mathematics, General". The system should treat these as equivalent to common shorter names, e.g., "English" and "Mathematics".
- name: a substring of the college's name. Note: in this context, the substring can match multiple colleges; disambiguation is unnecessary.
- ranking: range for the college's ranking.
- size: range for number of undergraduate students. The system gets that number from College Scorecard field UG if it is non-null otherwise from College Scorecard field UGDS.
- SAT Math, SAT EBRW, ACT Composite: range for average scores for enrolled freshmen. The systems gets this information from CollegeData.com; if CollegeData.com doesn't report the average, and it does report the range of middle 50%, use the midpoint of that range.
- Although the original source of each data item is listed above, this function uses the copy of that data in the system's database.

2.1.4 College Recommender

When viewing results of a college search, the student can ask the system to compute a recommendation score for each college in the search results, and sort the list by that score. The recommendation is based on where students with similar profiles applied and possibly other information. The student can select a college in the list to see the profiles of those similar students (to see how similar they are). Recommendation scores are computed upon request, rather than by default, because the computation is relatively expensive.

2.1.5 Find Similar High Schools

To get the most meaningful results from the applications tracker, a user needs to identify high schools similar to his/her own. This function helps the user do that. The user specifies a high school, and the system displays a list of the most similar high schools among those known to the system, with relevant information about them, sorted in descending order by similarity. [2020-01-31: updated the previous sentence.] The similarity metric should be based on a variety of relevant information about the academic quality of the high school, including information from niche.com (possibly including but not limited to the school's ranking) and information from this system (for example, standardized test scores or acceptance decisions in profiles of students from the high school).

2.1.6 Applications Tracker (a.k.a. Admissions Tracker)

For a specified college, the system displays information about other students who applied to the specified college and meet specified filter conditions. The user can specify whether to use strict or lax filtering. Supported filters include:

- college class: range of years of expected college graduation.
- high schools: a list of specified high schools. [2020-01-31: updated this item.]
- application status: a subset of the possible statuses.

Information about matching student profiles may be displayed in two forms.

2.1.6.1 List of Matching Student Profiles

List is sorted by application status, with key information from each, and a statistical summary of the listed profiles, specifically, average GPA, average SAT Math, average SAT EBRW, and average ACT composite, reported for all matching users, and for matching users with application status = accepted. The user can select a student profile to see all information in it.

2.1.6.2 Scatterplot of Application Status of Matching Student Profiles

Each matching student is represented by a point on the scatterplot. The color of the point indicates the student's application status; for example, green = accepted, red = denied, yellow = other. The mean values on the horizontal and vertical axes are shown by a dashed vertical line and a dashed horizontal line, respectively. The vertical axis is GPA. The horizontal axis is based on standardized test scores. The user can choose between SAT (Math+EBRW), ACT Composite, or weighted average of percentile scores for standardized tests (except AP tests). The weights used in the weighted average are: 5% for each SAT subject test taken, and the remainder for SAT or ACT Composite (or split evenly between SAT and ACT Composite, if the student took both).

2.2 Functionality for Administrators

All scrape and import functions overwrite old information in the system's database, if any.

2.2.1 Scrape College Rankings

Scrape WSJ/THE 2020 rankings of all colleges in colleges.txt.

2.2.2 Import College Scorecard Data File

Import information about all colleges in colleges.txt.

2.2.3 Scrape CollegeData.com

Scrape information about all colleges in colleges.txt.

2.2.4 Delete All Student Profiles

2.2.5 Import Student Profile Dataset

2.2.6 Review Questionable Acceptance Decisions

The system displays student profiles with questionable acceptance decisions. The administrator may validate some questionable acceptance decisions, so that they are no longer marked as questionable.

2.3 Use Case Diagram

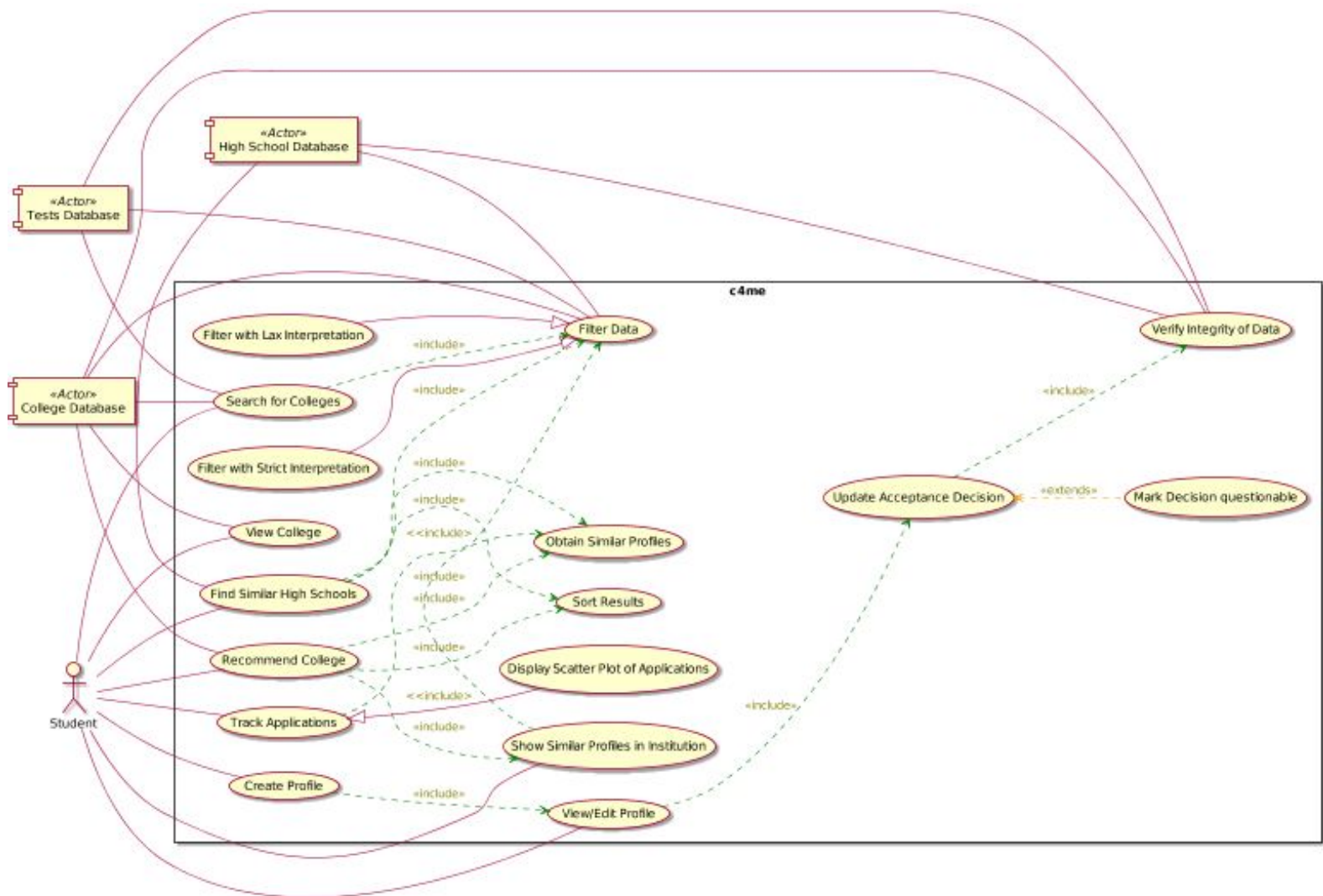


Figure 2.3.1: Student Use Case Diagram.

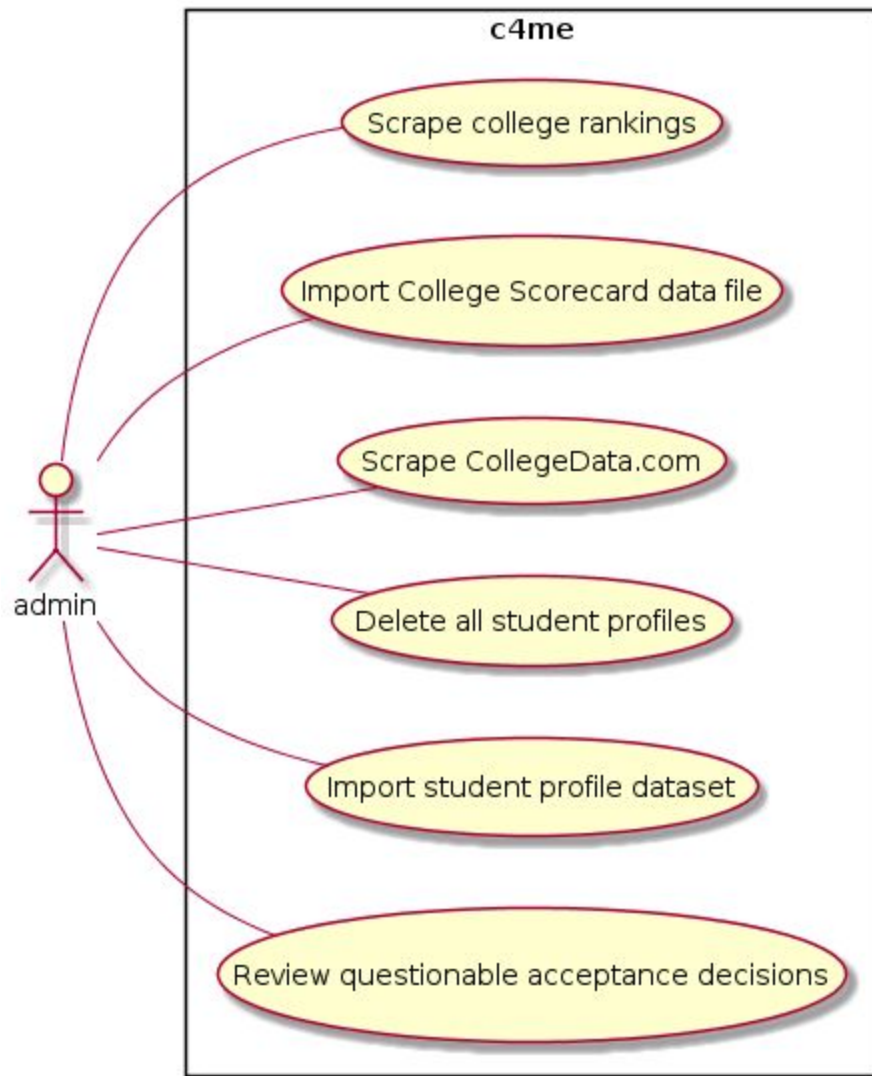


Figure 2.3.2: Admin Use Case Diagram.

2.4 Use Cases

Figure 1: List of Use Cases

Use Cases for Students	
2.4.1	Create Account

2.4.2	View/Edit Profile
2.4.3	Update Acceptance Decision
2.4.4	Verify Integrity of Data
2.4.5	Filter Data
2.4.6	Filter with Lax Interpretation
2.4.7	Filter with Strict Interpretation
2.4.8	Sort Results
2.4.9	View College
2.4.10	Search for Colleges
2.4.11	Recommend Colleges
2.4.12	Show Similar Profiles in Institution
2.4.13	Obtain Similar Profiles
2.4.14	Find Similar High Schools
2.4.15	Track Applications
2.4.16	Display Scatterplot of Applications
Use Cases for Administrators	
2.4.17	Scrape College Rankings
2.4.18	Import College Scorecard Data file
2.4.19	Scrape CollegeData.com
2.4.20	Delete All Student Profiles
2.4.21	Import Student Profile Dataset
2.4.22	Review Questionable Acceptance Decisions

Use Case 2.4.1

Use Case:	Create Account
Brief Description:	Student is added to the student profile database
Primary Actor:	Student

Precondition:	Student is logged out
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System prompts student for name, password, and password confirmation 2. Student enters the name, password, and password confirmation 3. Student presses 'Create Profile' 4. System sends information to the 'Student Profile Database' 5. 'Student Profile Database' adds student 6. Include 'View/Edit Profile' use case
Alternative Flow:	Invalid Authentication 5.1 System sends invalid input message

Use Case 2.4.2

Use Case:	View/Edit Profile
Brief Description:	Student views or edits the profile page
Primary Actor:	Student
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System displays 'Edit Profile' page 2. Student optionally edits information such as SAT scores in the profile 3. Student optionally updates acceptance decision, include 'Update Acceptance Decision' use case 4. Student clicks on the "Save" button 5. System sends updated information to database

Use Case 2.4.3

Use Case:	Update Acceptance Decision
Brief Description:	Student enters an acceptance decision and the system checks for verification
Primary Actor:	Student
Precondition:	Student updates the Acceptance Decision for a college in the profile
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System collects the information from the Student's profile 2. System checks statistical consistency between that decision and the information collected 3. Include 'Verify Integrity of Data' use case 4. System displays "Upload Successful"

Alternative Flow:	Mark Decision Questionable 4.1 System fails to confirm that the acceptance decision is likely 4.2 System marks it as questionable and indicates this to the student
-------------------	---

Use Case 2.4.4

Use Case:	Verify Integrity of Data
Secondary Actors:	College, Tests, and High School Databases
Brief Description:	Questionable acceptance decisions are ignored by the system when computing all query results
Primary Actor:	Student
Precondition:	Student is viewing search results
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System goes through the result list and removes every questionable acceptance decision

Use Case 2.4.5

Use Case:	Filter Data
Brief Description:	Student can optionally choose to filter the results
Primary Actor:	Student
Secondary Actor:	College, Tests, and High School databases
Primary Flow of Events:	<ol style="list-style-type: none"> 1. Student optionally filters by admission rate, cost of attendance, location, major, etc 2. System returns all results matching filters

Use Case 2.4.6

Use Case:	Filter with Lax Interpretation
Brief Description:	System treats missing data as true

Primary Actor:	Student
Trigger:	Student asks for a Lax Interpretation to be applied on the search result
Precondition:	Student is viewing search results
Primary Flow of Events:	The primary flow of events is the same as for the 'Filter Data' use case, except step 2 is replaced with the following: 2. System returns all results matching filters including those with missing items

Use Case 2.4.7

Use Case:	Filter with Strict Interpretation
Brief Description:	System treats missing data as false
Primary Actor:	Student
Trigger:	Student asks for a Strict Interpretation to be applied on the search result
Precondition:	Student is viewing search results
Primary Flow of Events:	The primary flow of events is the same as for the 'Filter Data' use case, except step 2 is replaced with the following: 2. System returns all results matching filters except those with missing items

Use Case 2.4.8

Use Case:	Sort Results
Brief Description:	Student can optionally sort the results
Primary Actor:	Student
Primary Flow of Events:	1. Student can optionally sort a field in ascending / descending order

Use Case 2.4.9

Use Case:	View College
-----------	--------------

Brief Description:	Student views the detail of a college
Primary Actor:	Student
Secondary Actor:	College Database
Precondition:	System displays a list of colleges
Trigger:	Student select a college from the list
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System contacts the College Database to displays additional information for selected college 2. Student can repeat process as desired

Use Case 2.4.10

Use Case:	Search for College
Primary Actor:	Student
Secondary Actor:	College Database, Tests Database
Precondition:	Student is logged in
Primary Flow of Events:	<ol style="list-style-type: none"> 1. Student submits the search form 2. System connects to the databases to find relevant results 3. Include 'Filter Data' and 'Sort Results' use cases 4. System displays the search result 5. Student can go back to step 1 as desired

Use Case 2.4.11

Use Case:	Recommend Colleges
Primary Actor:	Student
Secondary Actor:	College Database
Trigger:	Students asks for a recommendation score
Precondition:	Student is viewing results of a college search
Primary Flow of Events:	<ol style="list-style-type: none"> 1. Include 'Obtain Similar Profiles' use case 2. For each college, system computes

	recommendation scores 3. Include 'Sort Results' use case 4. System displays sorted colleges 5. Include 'Show Similar Profiles in Institution' use case
Alternative Flow:	2.1 System sets default values if no information in Student's profile for the queries available

Use Case 2.4.12

Use Case:	Show Similar Profiles in Institution
Primary Actor:	Student
Trigger:	Student clicks on specific Institution
Precondition:	Student is viewing Institution sorted by recommendation score
Primary Flow of Events:	1. System retrieves information about Student's profile 2. Include 'Obtain Similar Profiles' use case 3. System displays result
Alternative Flow:	3.1 If no results retrieved, display 'No matches found'

Use Case 2.4.13

Use Case:	Obtain Similar Profiles
Primary Actor:	Student Profiles Database (in existing IT system)
Precondition:	Student is viewing Institution sorted by recommendation score
Primary Flow of Events:	1. System retrieves information about student's profile 2. System sends query to retrieve similar profiles 3. System returns the results

Use Case 2.4.14

Use Case:	Find Similar High Schools
Primary Actor:	Student
Secondary Actor:	High School Database
Precondition:	Student specifies a high school and ask for similar high schools
Primary Flow of Events:	<ol style="list-style-type: none"> 1. System retrieves information about the high school specified 2. Include 'Obtain Similar Profiles' use case 3. System contacts 'High School Database' to find similar high schools 4. Includes 'Sort Results' 5. System displays the result to the Student
Alternative Flow:	2.1 If no results retrieved, display 'No matches found'

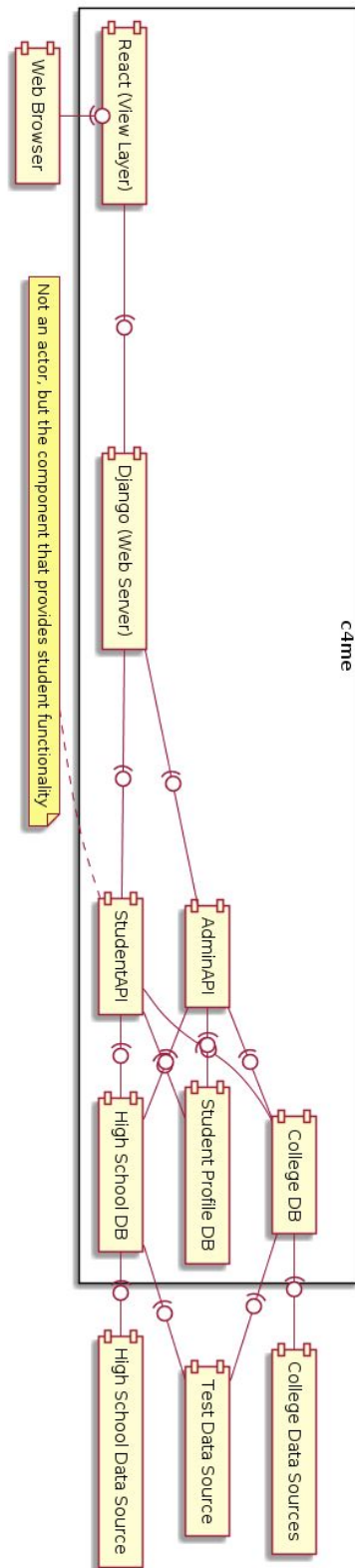
Use Case 2.4.15

Use Case:	Track Applications
Primary Actor:	Student
Precondition:	Student specifies a college and asks the System to display information about other students who applied to the specified college
Primary Flow of Events:	<ol style="list-style-type: none"> 1. The System retrieves the list of students that applied to the college 2. Include 'Filter Data' use case 3. Students optionally select to view a Scatterplot of the results

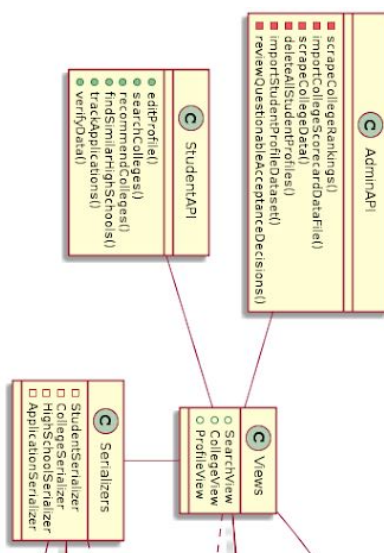
Use Case 2.4.16

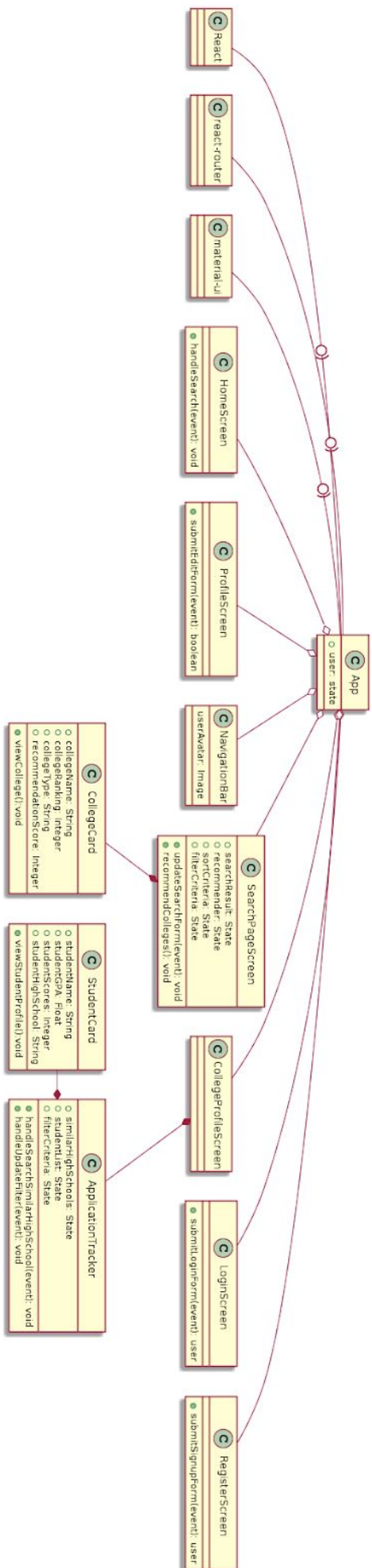
Use Case:	Display Scatterplot of Applications
Brief Description:	<p>The normal flow of events is the same as for the 'Track Applications' use case, except step 3 is replaced with:</p> <ol style="list-style-type: none"> 1. System returns all results in a scatterplot

2.6 Component Diagram



2.7 Class Diagram





2.8 Graphical User Interface and System Flow

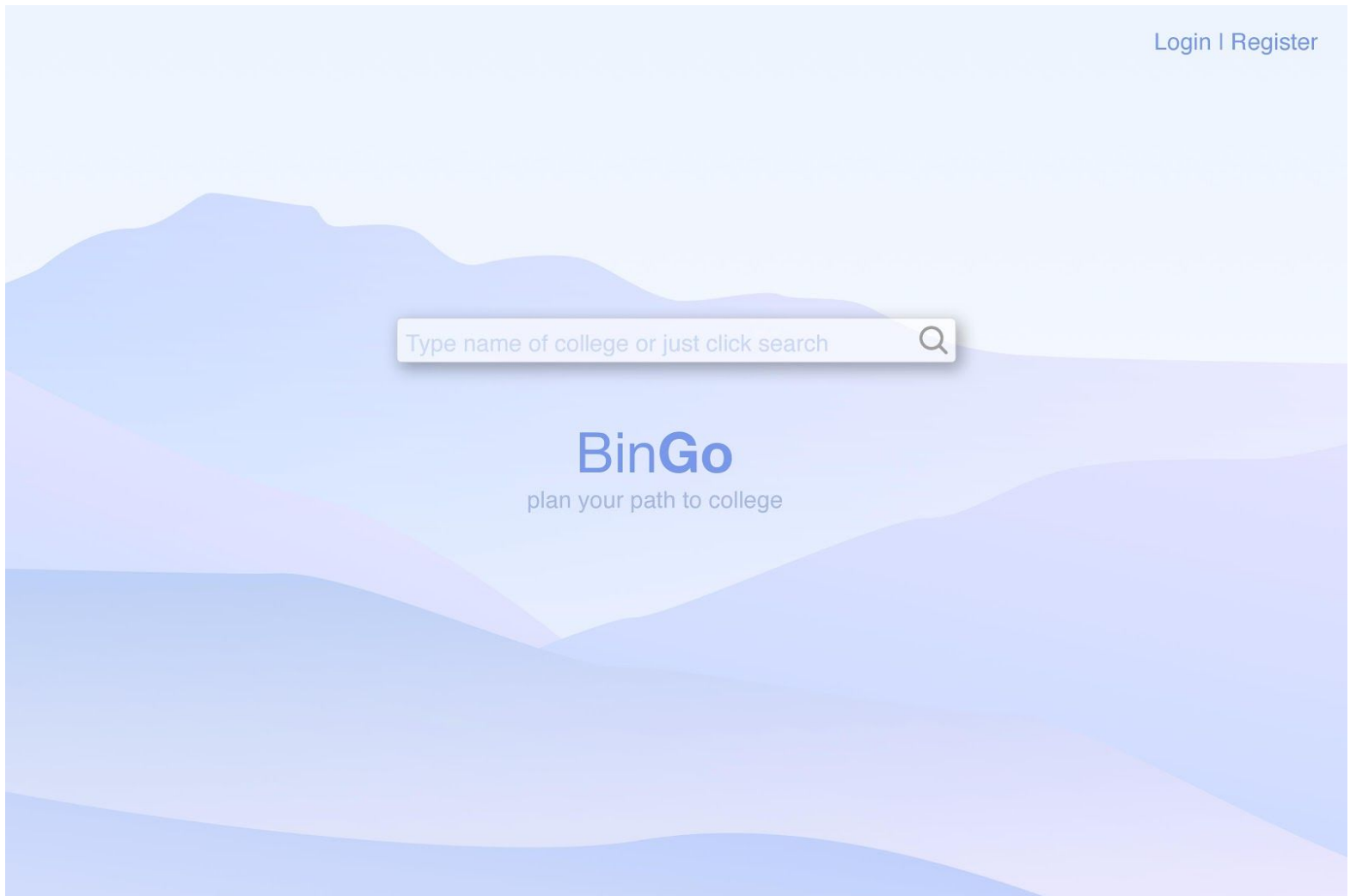


Figure 2.5.1: Home_Page

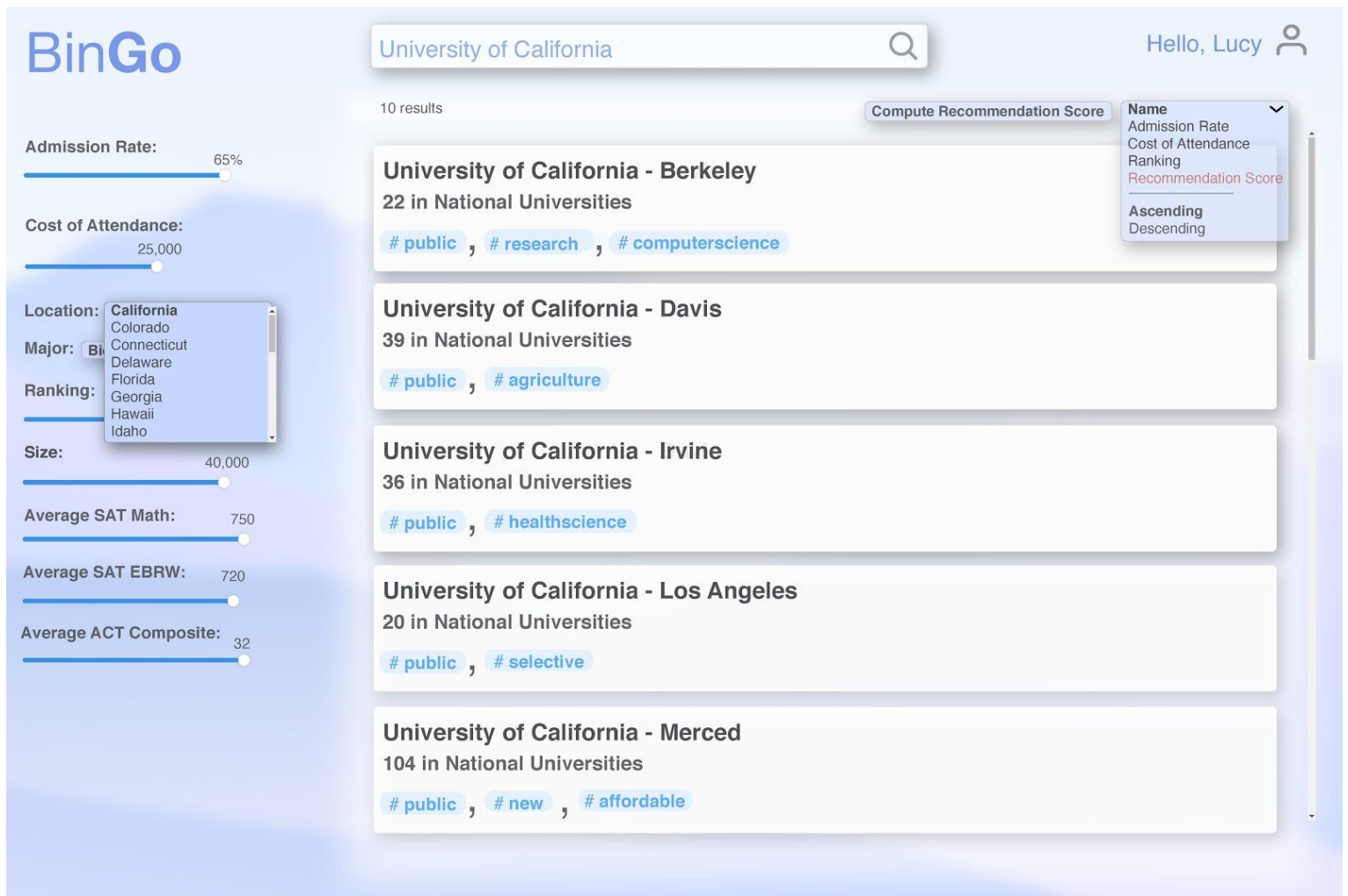


Figure 2.5.2: Search_Page

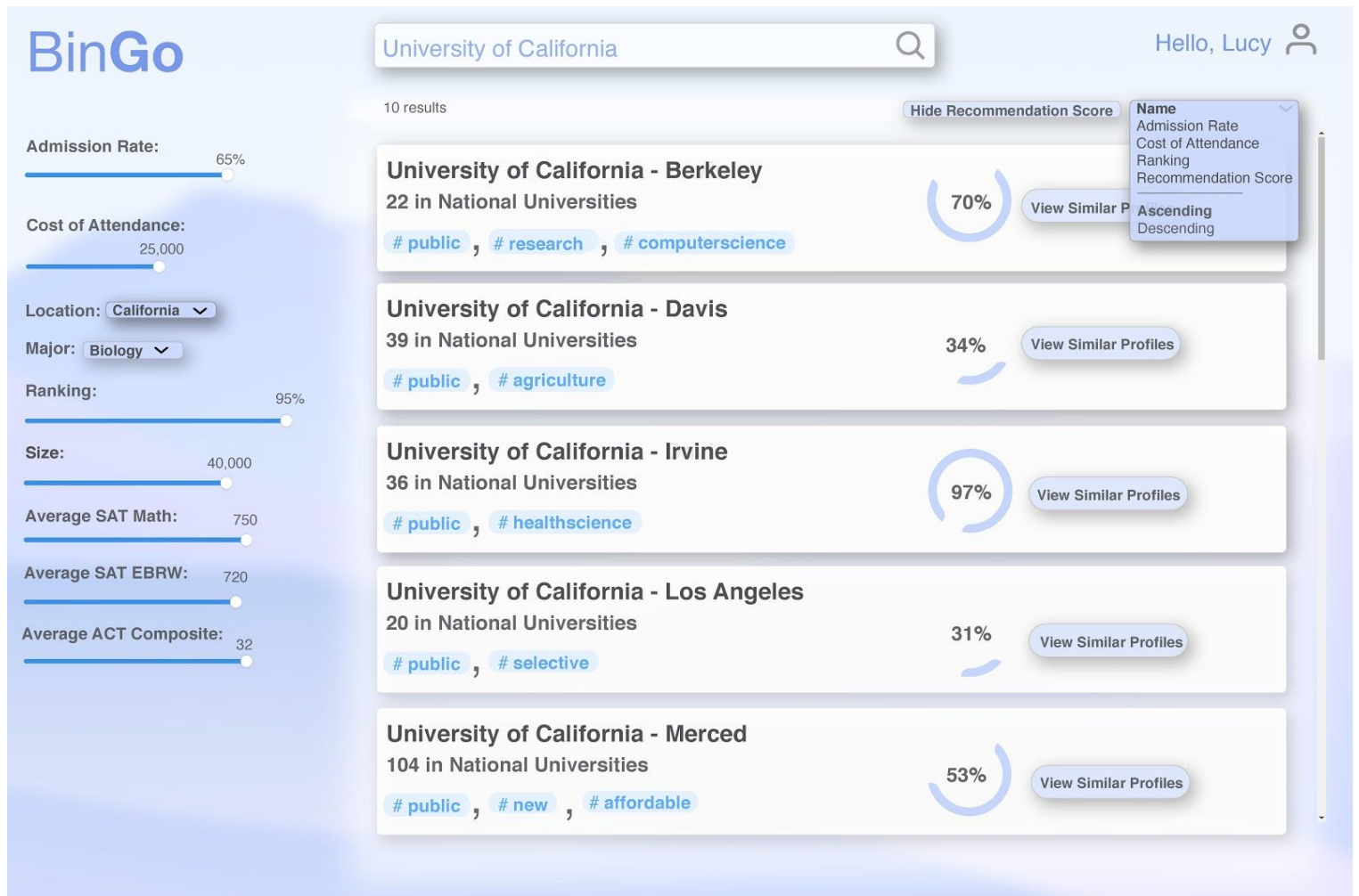



Figure 2.5.3: Recommendation_Page

#	High School Name	GPA	SAT Math	SAT EBRW	ACT Composite
1	Stuyvesant High School	3.96	—	—	35
2	Whitney High School	3.98	800	750	—

Figure 2.5.4: Recommendation_Profiles_Page

 Security Information Education**High School Name**

Stuyvesant High School ✓

High School City

New York City ✓

High School State

NY ▼

GPA

4.0 ✓

 Exam Scores College Applications

Save

Discard Changes

Figure 2.5.5: Edit_Profile

BinGo

Type name of college or just click search

Hello, Lucy

University of California - Berkeley

110 Sproul Hall, Berkeley, CA 94720 | (510) 642-6000

22 in National Universities

Quick Stats

Tuition and Fees (in-state)	\$14,184 (2019-2020)
Tuition and Fees (out-of-state)	\$43,176 (2019-2020)
Total Enrollment	42,501
Application Deadline	Nov. 30

Overview

Rankings

Admissions

Cost of Attendance

Map

Reviews

Here will be information about college.

Figure 2.5.6: View_College_Page

BinGo

Type name of college or just click search

Hello, Lucy

University of California - Berkeley

110 Sproul Hall, Berkeley, CA 94720 | (510) 642-6000

22 in National Universities

Quick Stats

Tuition and Fees (in-state)	\$14,184	(2019-2020)
Tuition and Fees (out-of-state)	\$43,176	(2019-2020)
Total Enrollment	42,501	
Application Deadline	Nov. 30	

Overview

Rankings

Admissions

Cost of Attendance

Map

Reviews

Type name of high school or just click search

Figure 2.5.7: View_College_Enter_HS_Page



University of California - Berkeley

110 Sproul Hall, Berkeley, CA 94720 | (510) 642-6000

22 in National Universities

Quick Stats

Tuition and Fees (in-state)	\$14,184 (2019-2020)
Tuition and Fees (out-of-state)	\$43,176 (2019-2020)
Total Enrollment	42,501
Application Deadline	Nov. 30

[Overview](#)[Rankings](#)[Admissions](#)[Cost of Attendance](#)[Map](#)[Reviews](#)

High School Entered:

Stuyvesant High
School

Brooklyn Technical High School

4 in New York City

public



Staten Island Technical High School

5 in New York City

public



Princeton High School

2 in Princeton

private



Next

Figure 2.5.8: View_College_Select_HS_Page

University of California - Berkeley

110 Sproul Hall, Berkeley, CA 94720 | (510) 642-6000

22 in National Universities

Quick Stats

Tuition and Fees (in-state)	\$14,184 (2019-2020)
Tuition and Fees (out-of-state)	\$43,176 (2019-2020)
Total Enrollment	42,501
Application Deadline	Nov. 30

Overview

Rankings

Admissions

Cost of Attendance

Map

Reviews

College Class:

2023 

Application Status:

Accepted 

High Schools:

Stuyvesant
High School Princeton
High School Staten Island
High School Brooklyn
Technical High
School [Change High
School List](#)

#	High School Name	GPA	SAT Math	SAT EBRW	College Class
1	Princeton High School	4.0	800	790	2023
2	Brooklyn Technical High School	3.98	800	780	2023

Figure 2.5.9: View_College_Admission_Tracker_Page

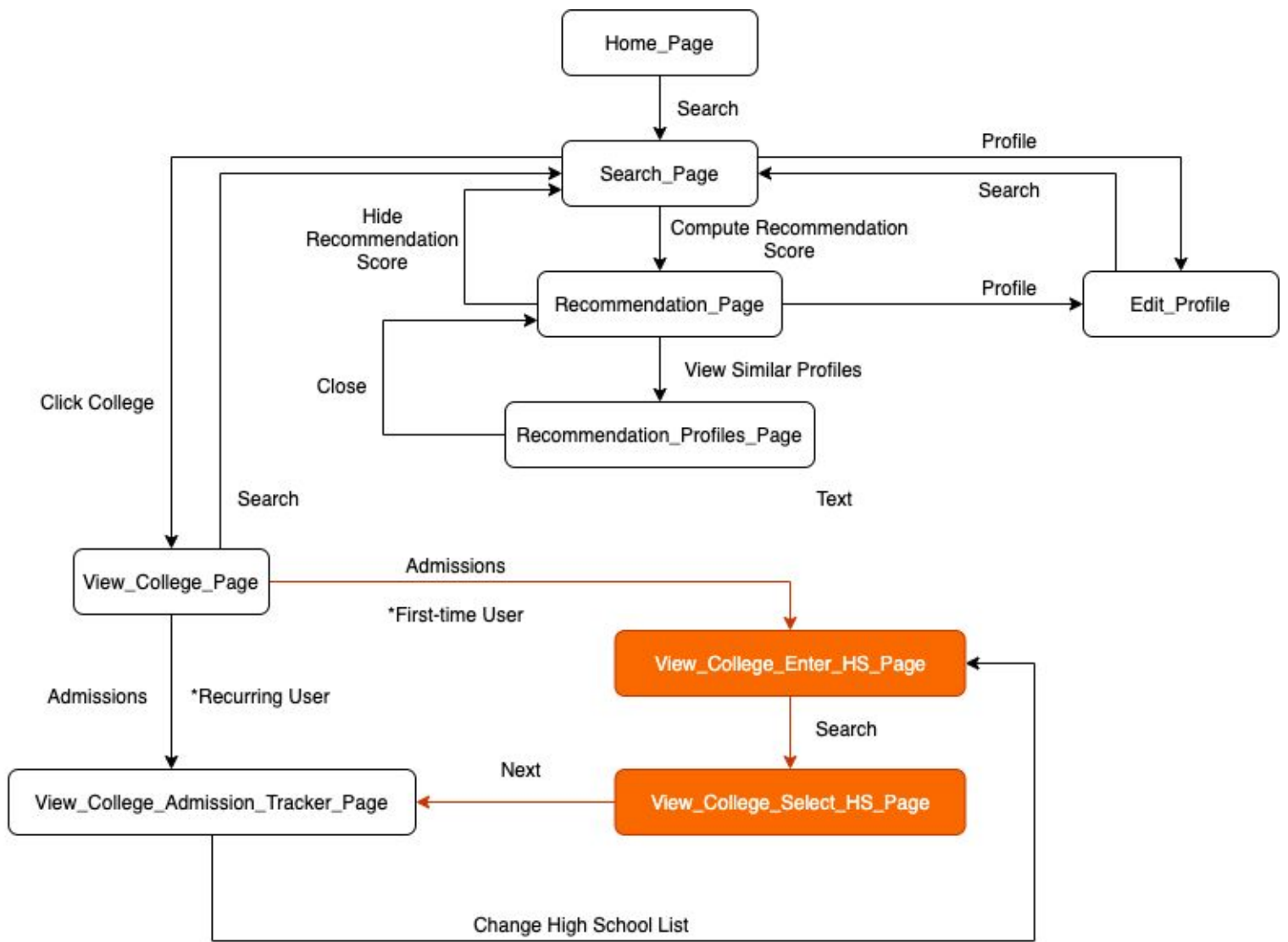


Figure 2.5.10: Window Navigation Diagram

2.9 Languages, Technologies, and Tools

Product	Tool
UML Diagram	PlantUML
GUI	Sketch
Window Navigation Diagram	Draw.io

Table 2.6.1: Tools For Design

Product	Tool
Frontend	React
Backend	Python, Django
Database	MySQL

Table 2.6.2: Tools For Implementation

3. Other Requirements

3.1 Authentication

All access to the system, [2020-01-30 added following phrase] except for student account creation, requires authentication with a password.

Passwords stored in the system database are salted and hashed. It is sufficient to have one pre-defined administrator account, with a fixed username and password. The system does not need to support the creation of additional administrator accounts.

3.2 Concurrency

Synchronization is used to ensure correct behavior when multiple users access the system concurrently.

3.3 Configuration

It should be easy (e.g., by editing one line in one file) to change the hostname and path prefix of URLs for all data source websites, for example, to change from <https://CollegeData.com/> to <https://www.cs.stonybrook.edu/CollegeData/>. I might create partial mirrors of some data source websites, to ensure availability and consistency of the content during demos.

3.4 Ease of Use

The system provides an easy-to-use, user-friendly web interface consistent with established UI design principles. The system handles invalid inputs gracefully, provides helpful feedback to the user when appropriate, etc.

3.5 Multi-host Operation

The client and server can run on different hosts.

3.6 Network Security

Network communication is secured using HTTPS or SSL. If your server does not have a public-key certificate signed by a certification authority trusted by the web browser, the web browser will show a security warning. The warning can be eliminated by creating a self-signed certificate, and installing the key in the browser, but that is optional. Requiring the user to tell the browser to proceed despite the security warning is acceptable.

3.7 Robustness

The system handles missing information gracefully. For example, almost all information in a student profile is optional (students should omit information they don't want to share, since all the information in the profile is public), and some information about colleges is missing on CollegeData.com (it's shown as "Not reported"). The system handles unreachable data sources gracefully, for example, if CollegeData.com is unreachable when scraping, or niche.com is unreachable when querying high school information. The system is robust to changes in the layout of scraped web pages, provided the part of direct interest is unchanged. The system can assume College Scorecard data files contain valid data. The system thoroughly validates data entered by students: high schools, colleges, test scores, etc. [2020-01-31: added the previous 3 sentences.]

4. Supporting Information

Note that this document should serve as a reference for designers and engineers in the future stages of the development process, so we will provide a table of contents to help quickly identify important sections.

4.1 Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Definitions, acronyms, and abbreviations
 - 1.4 References
 - 1.5 Overview
2. Overall Description
 - 2.1 Functionality for Students
 - 2.2 Functionality for Administrators
 - 2.3 Use Case Diagram
 - 2.4 Use Cases
 - 2.5 Component Diagram
 - 2.6 Class Diagram
 - 2.7 Graphical User Interface and System Flow
 - 2.8 Languages, Technologies, and Tools
3. Other Requirements
 - 3.1 Authentication
 - 3.2 Concurrency
 - 3.3 Configuration
 - 3.4 Ease of Use
 - 3.5 Multi-host Operation
 - 3.6 Network Security
 - 3.7 Robustness

4. Supporting Information

4.1 Table of Contents

4.2 Contributions

4.2 Contributions

HW2

Olesia: Use Cases and overview of the document

Omar: Use Cases and Use Case Diagram

Jafree: Use Cases and grammar

Bing: Use Cases and formats

Together: Designed and reviewed the document

HW3

Olesia: Graphical User Interface and System Flow

Omar: Backend Class Model

Jafree: Component UML Model

Bing: Frontend Class Model

Together: Designed UI and reviewed all the document