Milestone One

SW Engineering CSC648/848 Spring 2019

Team 04

02/24/2020

SFStateEats

Team Members

Name	Email	Role
Rachit Joshi	rjoshi@mail.sfsu.edu	Team Lead
Samhita Brigeda	pandu.barigeda@gmail.com	Front End Lead
Pedro Souto	Pedro.Souto.SFSU@gmail.com	Back End Lead
John Pham	JohnPhamDeveloper@hotmail.com	Github Master
Vincent Tran	vtran6@mail.sfsu.edu	Database Lead
Khang Tran	ktran26@mail.sfsu.edu	Frontend Developer

Milestone 1 Version History

Version Name	Date
Milestone 1 version 1	02/24/2020

Table of Contents

Section 1: Executive Summary

Currently, San Francisco State does not have an organized and easy method to view food locations on campus. There are many applications that are used to find great restaurants and popular food spots. For example, *Yelp* and *Google Restaurants* are greatly used outside of campus and for personal use. On our campus, there is a lack of information on smaller restaurants and food locations. Some food locations like cafes, restaurants, and food courts are too small to show up on some of the previously mentioned services. ***SFStateEats*** will be a service that allows its users to rate, view, and discover all food places on San Francisco State University Campus. It will also allow restaurant owners to add their restaurants to the platform.

Fact is, trends catch on. New and hot apps spread between users, and what better place to spread a new service than a close knit school campus? Our service will increase in users via a network effect, as more and more students begin to experience SFStateEats. The reach that this web application can have is enormous, with San Francisco State University being just the tip of the iceberg.

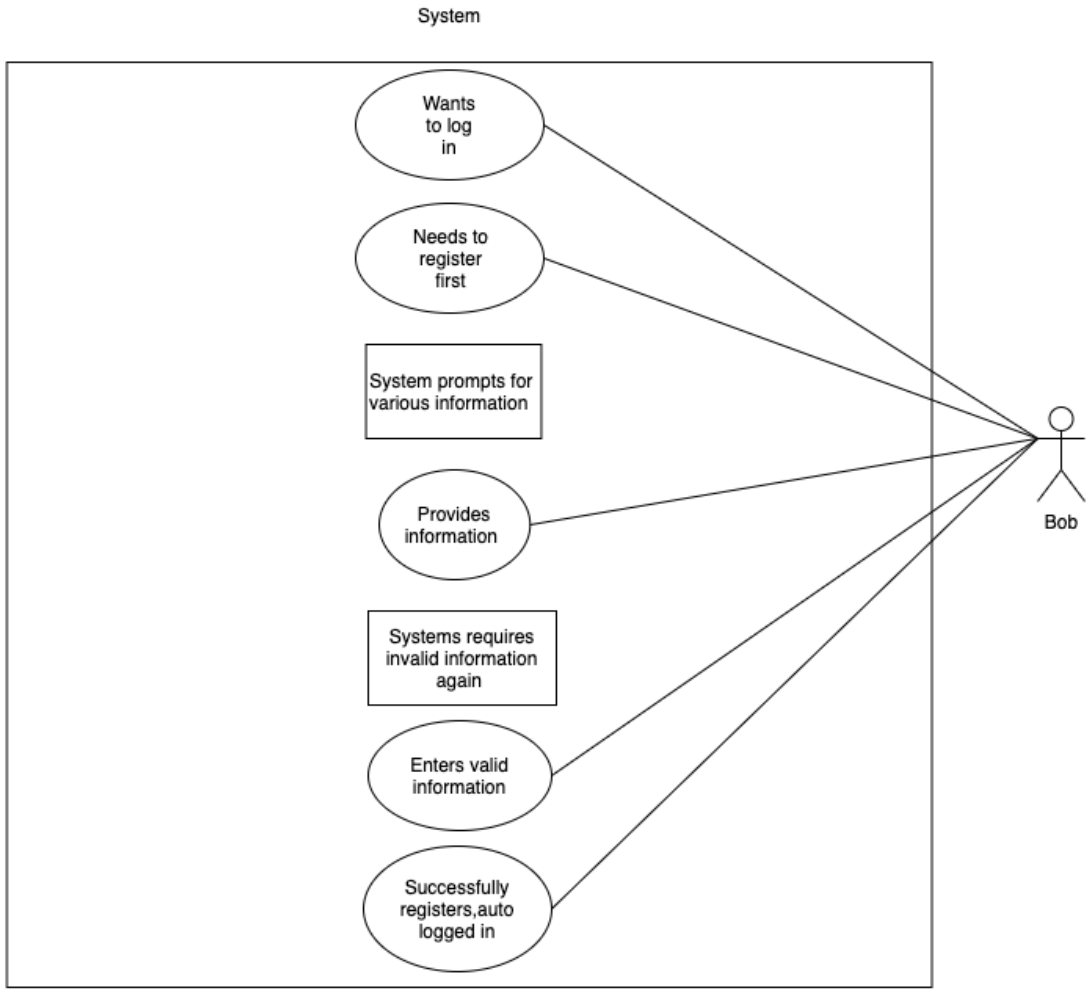
We plan to launch our web application in the beginning of summer 2020, where every student of San Francisco State University will be able to join us. When this happens, all features will go live.

"Our mission at SFStateEats is to make sure ***every*** *student has access to* ***all*** *information of* ***every*** *restaurant on campus."*

Section 2: Main Use Cases

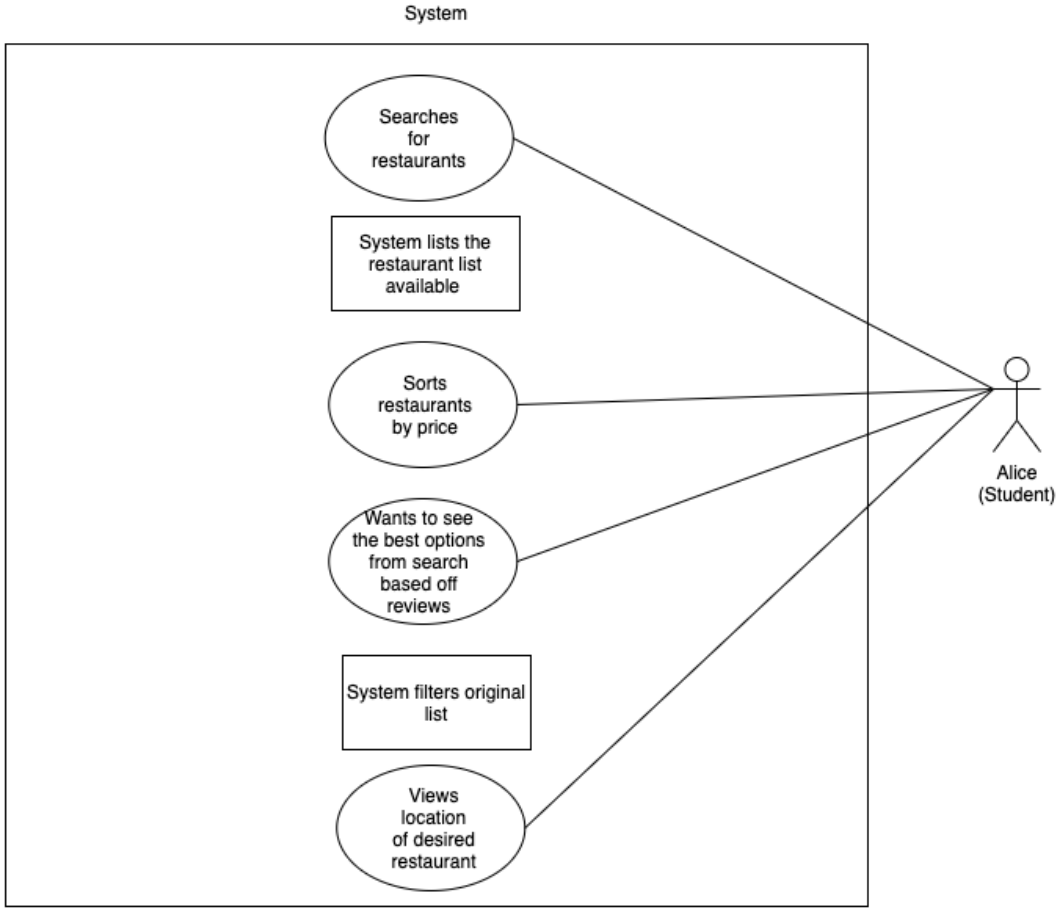
1. Bob just visited Caffe Rosso on campus by himself and had an amazing breakfast sandwich. Bob opens the SFStateEats application to give the restaurant a 5 star review. The SFStateEats application first asks Bob to login before he can rate a restaurant, but Bob does not have an account. Bob attempts to register for a regular user account, but the username he's trying to register with is already taken. Bob is given a prompt to try a different username instead. Once Bob has finished the registering process, he is given a message stating his registration is successful and is automatically logged in.

Use case 1:

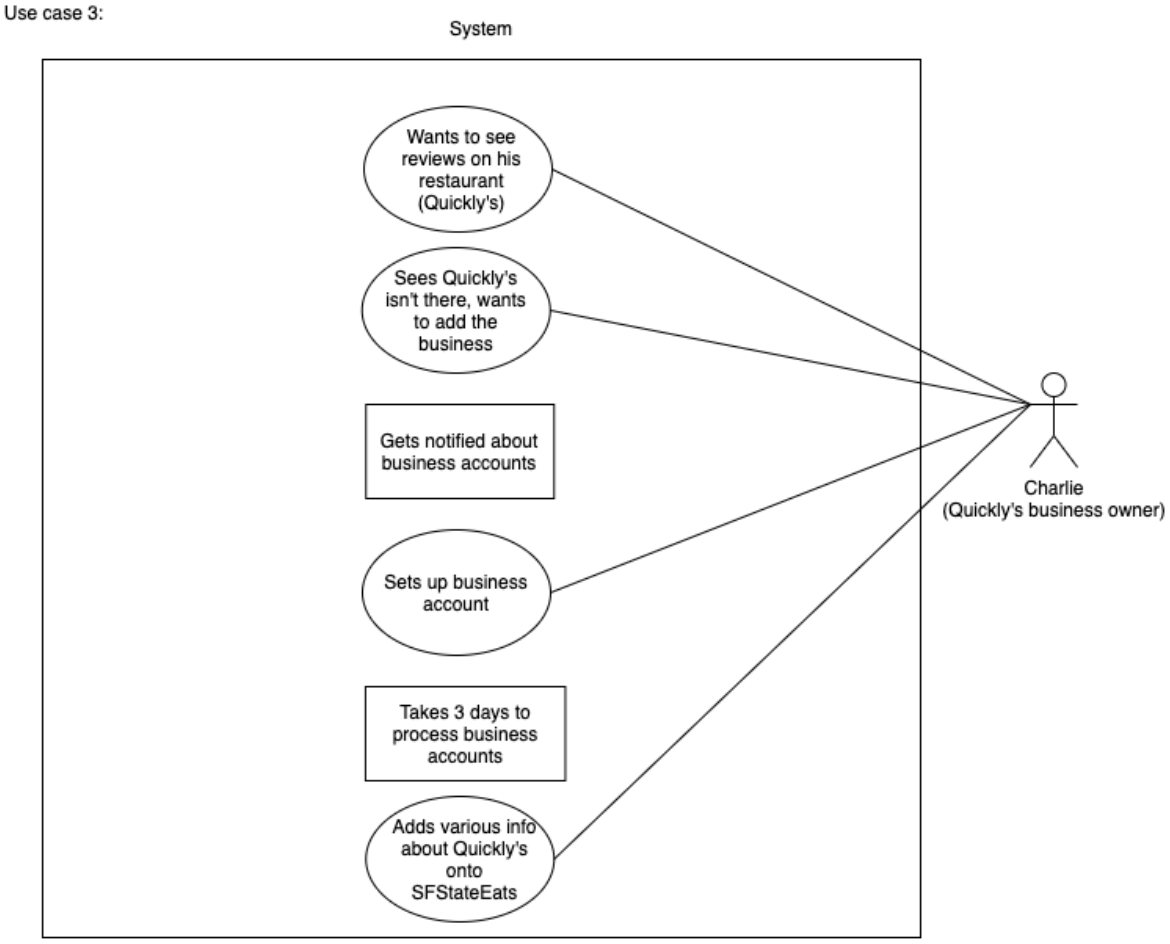


2. Alice is on a budget and wants to find a cheap place to eat on campus before her next class. She opens the SFStateEats application and searches for every food place on campus. Since she wants the cheapest options, she sorts the list by price, then looks for a location with good reviews. She sees that Nizario's Pizza is located in the basement of the Cesar Chavez Student Center building. She also sees that this food place has 4.5 / 5 stars, and decides to try it out.

Use case 2:

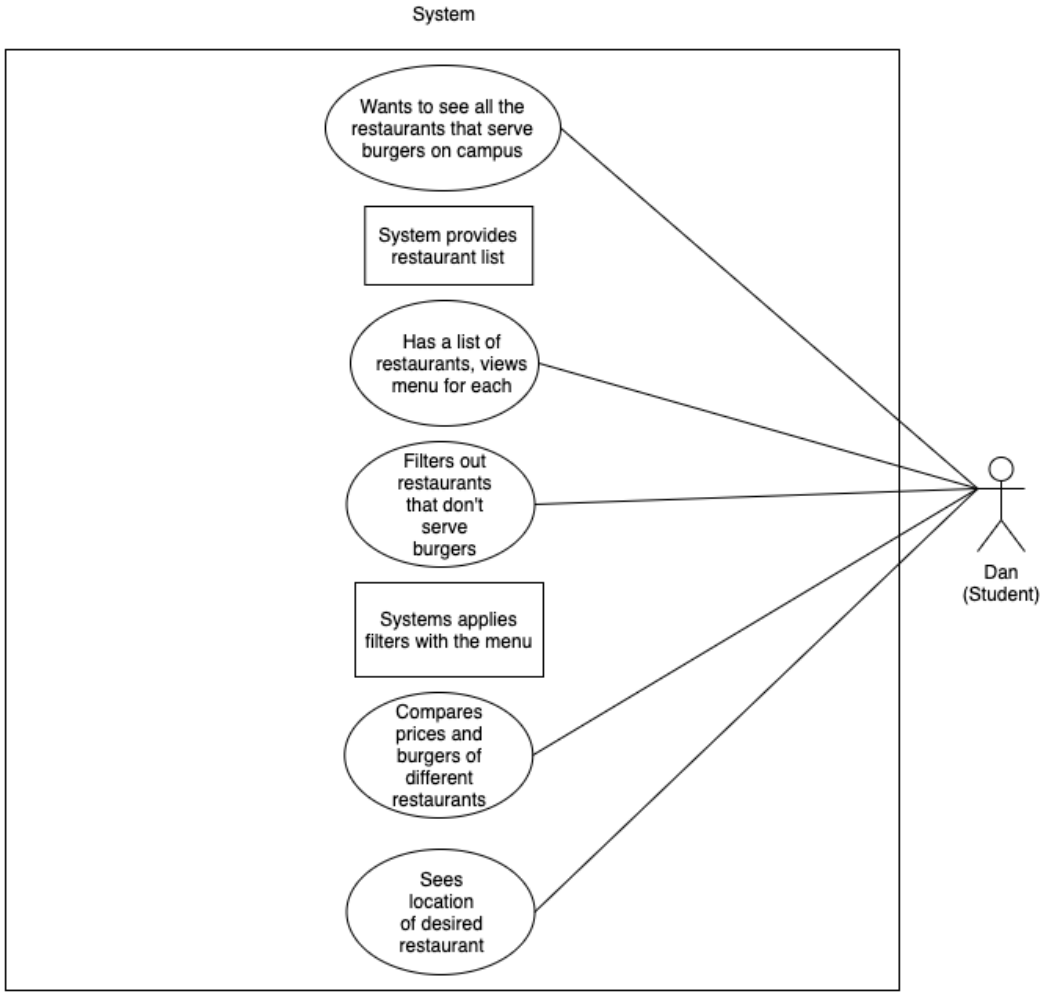


3. Charlie is the owner of a food place called Quickly's on campus. He heard about this great web app called SFStateEats and wants to see what his customers think of his food. He looks up Quickly's on SFStateEats, and realizes this restaurant is not listed on the web app. He tries to add his restaurant onto the platform, but is notified that he needs a business account in order to add a new place to the platform. He starts the process to upgrade his regular account to a business account, and after 3 days, it is verified. With his business account, he adds Quickly's to the list of campus restaurants, along with the menu, hours, and prices.



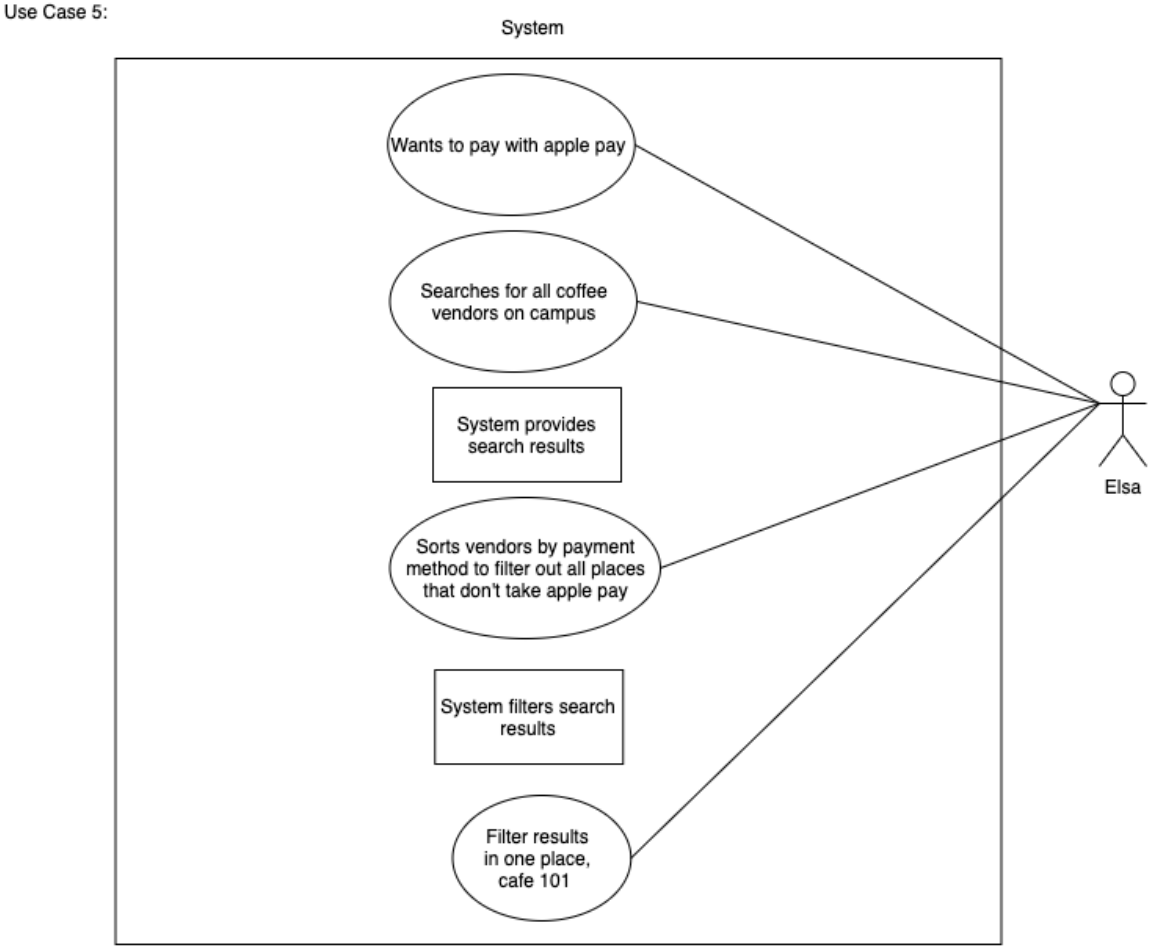
4. Dan is a student at San Francisco State University and loves trying out new food. He is looking for a good burger for lunch today and wants to see all the burgers on campus. He opens the list of all restaurants, and pulls up the menu for each of them. He then filters out all the menus that do not have a burger as an item on it, and is left with a handful of menus that do. He compares the prices and description for all the burgers, and decides to go with the burger on the third menu. He is able to see that it belongs to a place called Burger Express, located on the first floor of the Ceasar Chavez building.

Use case 4:

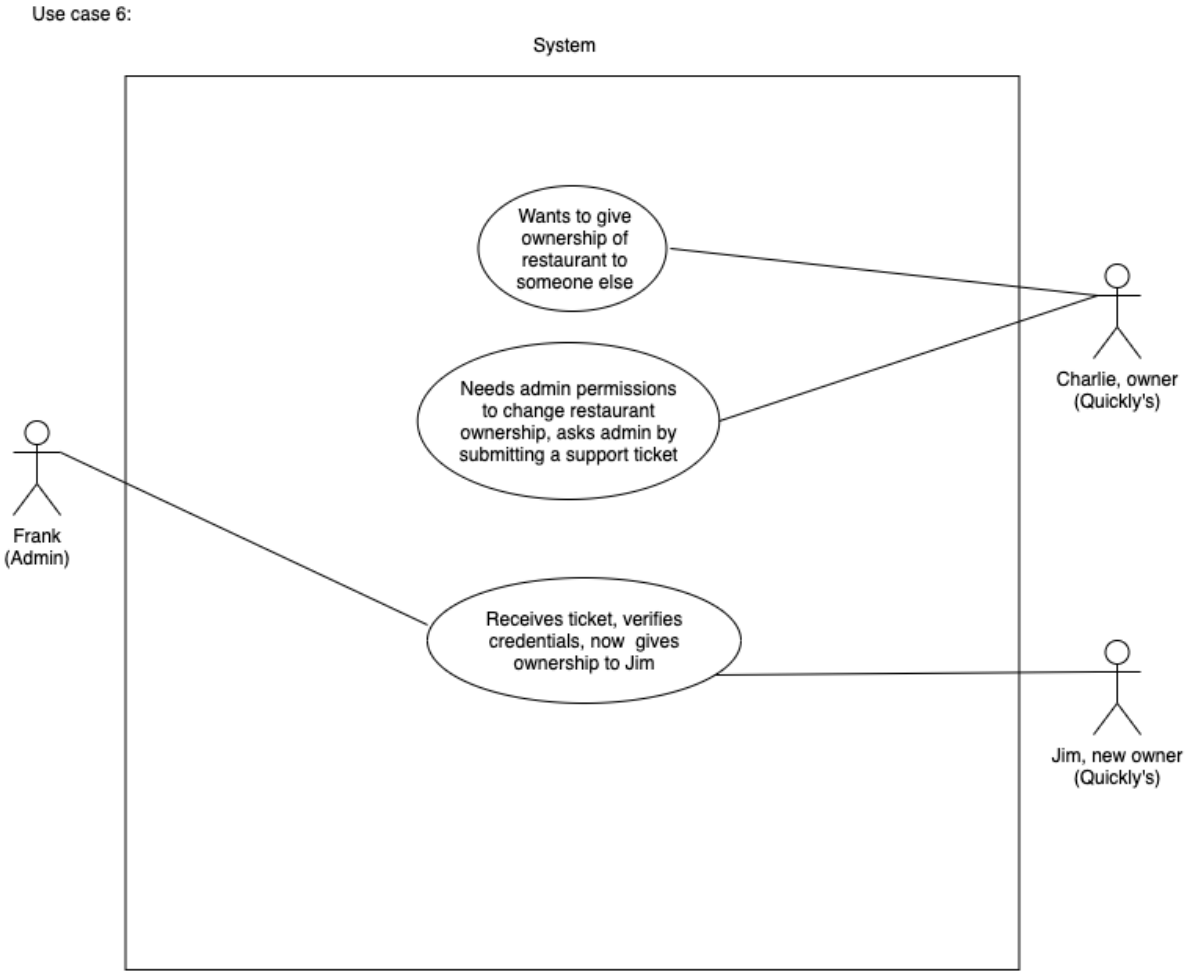


5. Elsa is coming early to campus to meet with a study group. In her rush to get out of the door in the morning, she forgot her wallet at home, only has her iPhone. She needs a coffee to get her day started, but does not have any cash or credit cards with her. She opens SFStateEats, and pulls up all the vendors that have coffee. She then sorts it by payment method to filter out all the places that do not accept ApplePay. Luckly, one place did, so she decides to go to Cafe 101.

Use Case 5:

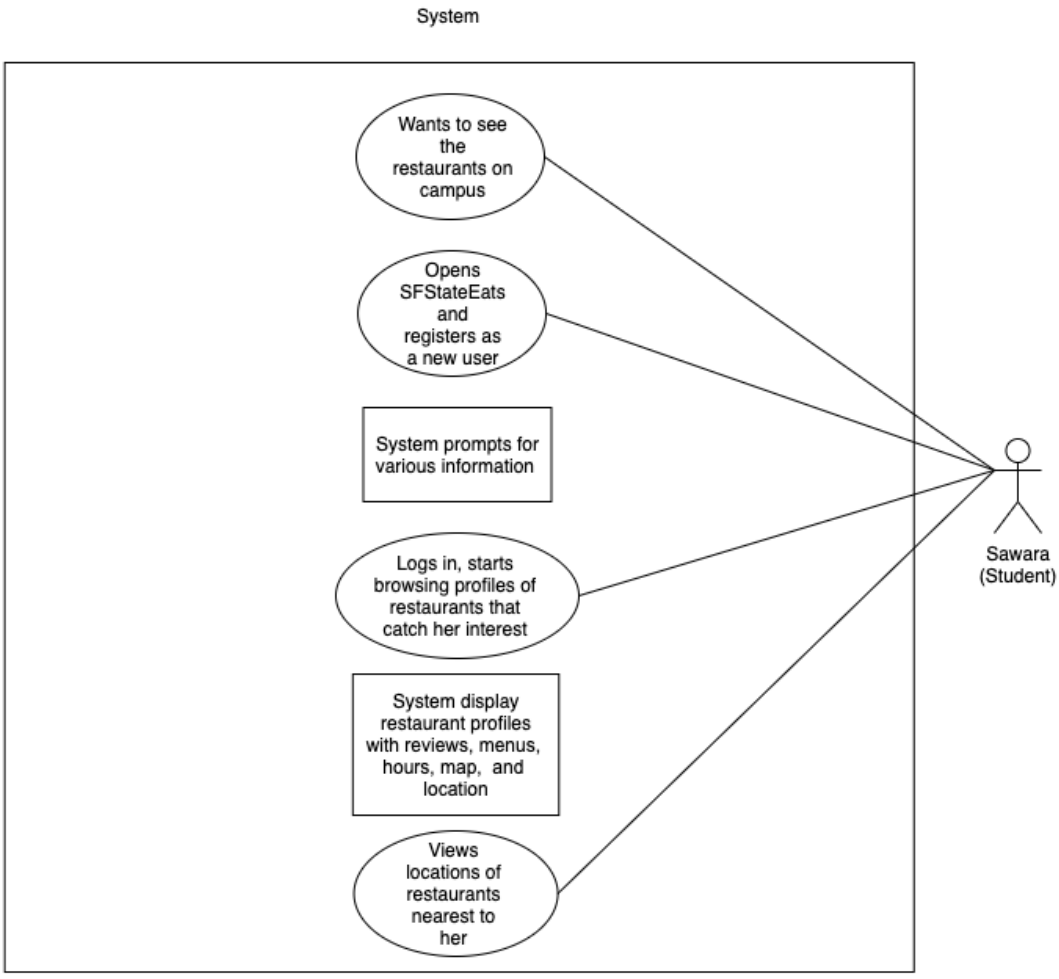


6. Frank is an admin on the SFStateEats app. When users encounter a problem on the app, they can open a support ticket to receive help. Charlie, who is the owner of Quickly's is selling his restaurant to move to another state. He needs to change the owner on the Quickly's account to someone else, so he opens a support ticket to have an admin do it. Frank receives a support request, and after verifying all the proper paperwork, Frank manually changes the owner of Quickly's to the new owner, Jim.

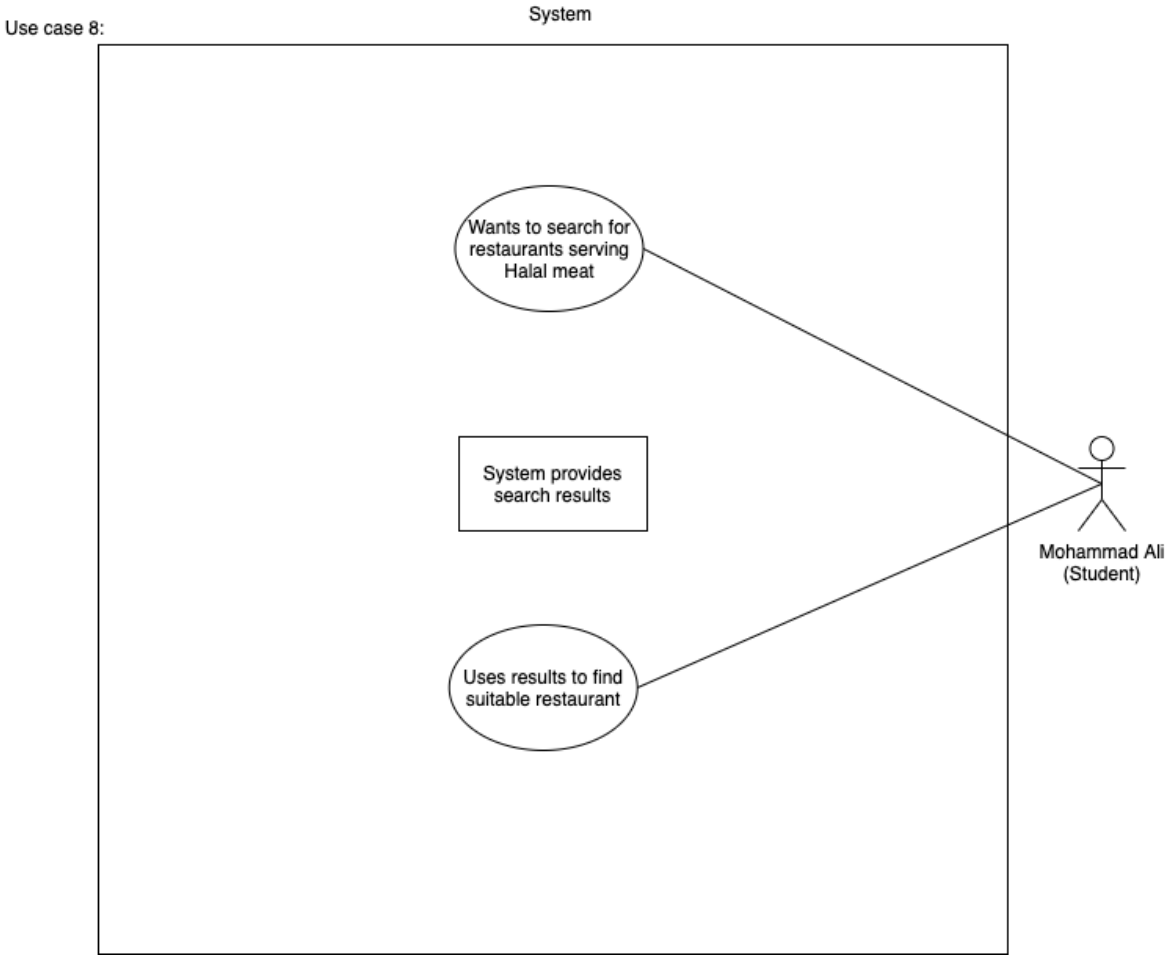


7. Sawara is a new student at SF State, and she wants to look at the food places on campus. She learns about an app called SFStateEats that is for all the restaurants on campus. She downloads the app and registers for a new account. After registering, she logs into the app and starts exploring the profile pages of all the restaurants that she is interested in. Then she starts seeing the locations on the map to see which is closest to her. Based on her search and interests, she choses to get an item at Cafe Russo.

Use case 7:



8. Mohammad Ali is a current student at SFSU. He wants to know if there are any restaurants on campus that serve halal meat. Mohammad logs into SFStateEats and uses halal meat in his search. After skimming through the list of restaurants serving halal meat, he decides to go to Crave to get a sub.



Section 3: List of Main Data Items and Entities

Entity or Data Item	Description
System Administrator	Account for those users who are employed or trusted by the application owners. These accounts have access to moderator tools that no other account has access to. This may include the ability to remove reviews or change business information.
Unregistered User	A user who visits the website to use its features, but does not create an account.
Registered User	A user who uses the website and has created an account. This allows them to access features that an unregistered user may not have.
Registered Business Owner	A user who owns a business that is listed on SFStateEats. This account has access to features relating to business pages that other users do not.
Restaurant	An entity that contains all the information about a restaurant. This will be the page that has reviews, menus, hours, map, and location.
Review	A data item relating to restaurants. Reviews can be left by registered users and can be seen by any user. Reviews may include a section of text, photos, and ratings.
Menu	A data item relating to restaurants. A menu lists all of the items that the restaurant servers. This may include prices along with other information.
Map/Location/Directions	A data item relating to restaurants. A location will be information regarding where the restaurant can be found.

Section 4: Initial List of Functional Requirements

User

1. New users shall be able to register a customer account.
2. Users shall be able to login into their account.
3. Users shall be able to rate restaurants on campus.
4. Users shall be able to rate food places on campus.
5. Users shall be able to write and view reviews.
6. Users shall be able to reply to reviews.
7. Users should be able to post pictures in their reviews
8. Users shall be able to see the menu for each restaurant.
9. Users shall be able to see the price for each item on the menu.
10. Users should be able to see the location of the restaurant on the profile page.
11. Users should be able to see all the stalls/business that are involved in the farmers market.
12. Users should be able to see the map locations of each restaurant on campus.
13. Users and Business Owners should be able to add photos of each item on the website.
14. Users shall be able to check-in to restaurants.
15. Users should be able to see busy times for restaurants.
16. Users should be able to see hours of business for each restaurant.
17. Users should be able to apply filters to their search

Business

18. New business owners shall be able to register a business account.
19. Business owners shall be able to login into their account.
20. Business owners shall be able to add their restaurant to the website.
21. Business owners shall be able to create a menu for their restaurant.
22. Business owners should be able to upload photos of their restaurant.
23. Business owners should be able to tag their restaurant into categories.
24. Business owners should be able to write descriptions of their restaurant on the profile page.
25. Business owners shall be able to upload an approved health score to reassure their customers.

Restaurant

26. The restaurant profile page shall display all this information (photos, user ratings, tags, description, etc).

System Administrator

27. System Administrators should have the privilege to delete restaurants
28. System administrators should have the privilege to ban any users from the website for misuse.

Section 5: List of Non-Functional Requirements

Security

1. Login shall be required to leave reviews.
2. User’s shall verify their emails when registering for an account.
3. User’s shall be able to set a display name different from their email address.
4. User’s email shall not be displayed by default.
5. Passwords shall be encrypted before storing into database.
6. User’s sessions timeout limit shall be decided by the system administrator.
7. This site shall not accept any third party cookies.

Audit

8. New business account registrations shall be audited by a system administrator.
9. New business account registrations shall be approved by a system administrator.
10. Users shall not be able to log into system administrator accounts.
11. New business listing shall be approved by the system administrator.

Performance

12. Each service should be hosted in its own server to prevent overwhelming one server.
13. The web application should restart if it is abruptly shut down to prevent downtime.
14. System shall respond visually within 5 seconds for all screens.
15. Application shall be able to retrieve information from the database and react in a timely manner.
16. The site shall handle requests asynchronously following a REST format.

Capacity

17. The site shall be capable of handling at least 100 students.
18. The site shall be scalable, so that new features can be added easily.

Reliability

19. Downtime maintenance shall be less than or equal to 4 hours per month.
20. Downtime for maintenance shall not affect the site’s main functionality.
21. In all cases, users shall be informed of downtime for maintenance via an announcement on the main page.

Recovery

22. In case of a total site failure, the whole site shall be shut down for revision.
23. If the site is broken, the mean time to recovery shall not exceed one day.
24. User data is the most valuable aspect and priority will be placed on recovering such data in case of total failure.

Data Integrity

25. Database tables shall be backed up weekly.
26. System administrators shall be able to execute a recovery if needed.
27. Image sizes shall be restricted to at most 3 megabytes.
28. Images shall be uploaded in jpg, jpeg, or png formats.

Compatibility

29. The site shall be compatible with the latest version of the Safari browser.
30. The site shall be compatible with the latest version of the Firefox browser.
31. The site shall be compatible with the latest version of the Chrome browser.
32. Third party applications shall not be to modify any content that may affect the site compatibility.
33. Content should be able to be ignored by most popular ad-block services.
34. The site shall be able to account for any other compatibility issues created as a result of browser updates in the future.

Conformance with Coding Standards

35. Architecture and design shall meet all the requirements listed under the ‘High-level System Architecture and Technologies Used’ section of this document.
36. Design pattern is to be strictly enforced with all aspects of the site.
37. Appropriate documentation must be created for all code that is individually written for future maintenance.
38. Production code shall not have any log or output to the console.
39. All errors must not halt the web application without appropriate error handling.
40. Only working code that meets all coding standards shall be submitted to the main branch of the project repository.
41. Code shall be thoroughly tested and debugged before being considered working code.
42. All internal errors and exceptions encountered when writing or modifying code shall be stored in a log.
43. Any error that can affect the site’s functionality shall be reported to the user.
44. Errors shall be handled in a way that does not affect site functionality.
45. The site shall not be launched without all priority one features finished and working.
46. All major changes to the application shall be discussed by the team and communicated to the class CTO.

47. The whole production cycle of the site shall be finished in at least 5 days before the delivery date.
48. The site shall be tested and debugged as a whole product at least 3 days before the delivery date.

Look and Feel Standards

49. The application and it’s layouts shall look professional.
50. The site shall be simple, so that it is usable to a wide range of users, and all previously mentioned parties.
51. Targeted users will be the main priority for ensuring usability and readability.
52. Elements on screen shall meet the compatibility standards of all supported browsers.
53. Elements on screen shall meet the compatibility standards of all supported browsers on mobile devices.
54. Elements on screen shall be aesthetically pleasing.
55. The site shall be able to work correctly without mouse interaction.
56. The site shall be able to work correctly without keyboard interaction.
57. Elements on screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site.
58. Application’s user interface shall make it easy for users to find what they are looking for.

Internalization / Localization Requirements

59. The default language of the site shall be English.
60. The site shall only allow SFSU students to register accounts initially.
61. The site shall be scalable to incorporate other schools into the user base.

Web Site Policies

62. A link to the policies of this site shall always be visible in all its pages to be accessible by all the parties.
63. The website shall allow users to register for an account.
64. Email verification shall be implemented upon registration.
65. User’s shall agree to the application's privacy policy before using the product.

Section 6: Competitive Analysis

Feature/Company	Yelp	OpenTable	Google Reviews
Strengths	Sleek and intuitive UI	Unintrusive UI	More accurate restaurant information
Weaknesses	Restaurant information not always up to date	Commenting Section not fully fledged out	UI is not easy to navigate
Customer Support	Responsive customer support	Responsive customer support	Does not offer customer support
Featured Businesses	All businesses	Food businesses only	All businesses
Brand Recognition	Highly popular. Mildly accessible.	Less popular. Mildly accessible.	Mildly popular. Highly accessible
Onboard Experience	Fast and seamless browsing.	Small review base, but organized UI.	Large review base, but unorganized reviews.

Feature	Yelp	OpenTable	Google Reviews	SFStateEats
Reviews	+	+	+	++[1]
Reservation	+	+	-	-
Online Order	+	-	-	-
Campus Event Announcements	-	-	-	+[2]
Location and Hours	+	+	+	+

1: While other competitors also have reviews, none of them have threads within those reviews. SFStateEats shall have the ability to leave comments and start a thread on existing reviews. This is a benefit because people can discuss the topics mentioned in a specific review, as opposed to just the restaurant as a whole.

2: Yelp, OpenTable, and Google Reviews all lack the ability to add food related events that are occuring on campus. One example of such an event is when a conference occurs, they often give out free food. SFStateEats shall allow organizers of these events to announce their events.

Section 7: High Level System Architecture and Technologies Used

Server Host: Amazon EC2 server vCPUs 1 RAM 1GiB
Operating System: Ubuntu 18.04
Database: PostgreSQL v10.12
Web Server: EC2 t2.micro
Server-Side Language: JavaScript
Web Framework: ReactJS
IDE: IntelliJ, WebStorm, VS Code
HTML: 5
CSS: 3
React: 0.61
Node: 13.8.0
Supported browsers: Chrome, Safari, Firefox and supported engines.

Section 8: Team

Name	Email	Role
Rachit Joshi	rjoshi@mail.sfsu.edu	Team Lead
Samhita Brigeda	pandu.barigeda@gmail.com	Front End Lead
Pedro Souto	Pedro.Souto.SFSU@gmail.com	Back End Lead
John Pham	JohnPhamDeveloper@hotmail.com	Github Master
Vincent Tran	vtran6@mail.sfsu.edu	Database Lead
Khang Tran	ktran26@mail.sfsu.edu	Frontend Developer

Section 9: Checklist

Task	Status
Team found a time slot to meet outside of the class	DONE
Github master chosen	DONE
Team decided and agreed together on using the listed SW tools and deployment server	DONE
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	ON TRACK
Team lead ensured that all team members read the final M1 and agree/ understand it before submission	DONE
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	DONE