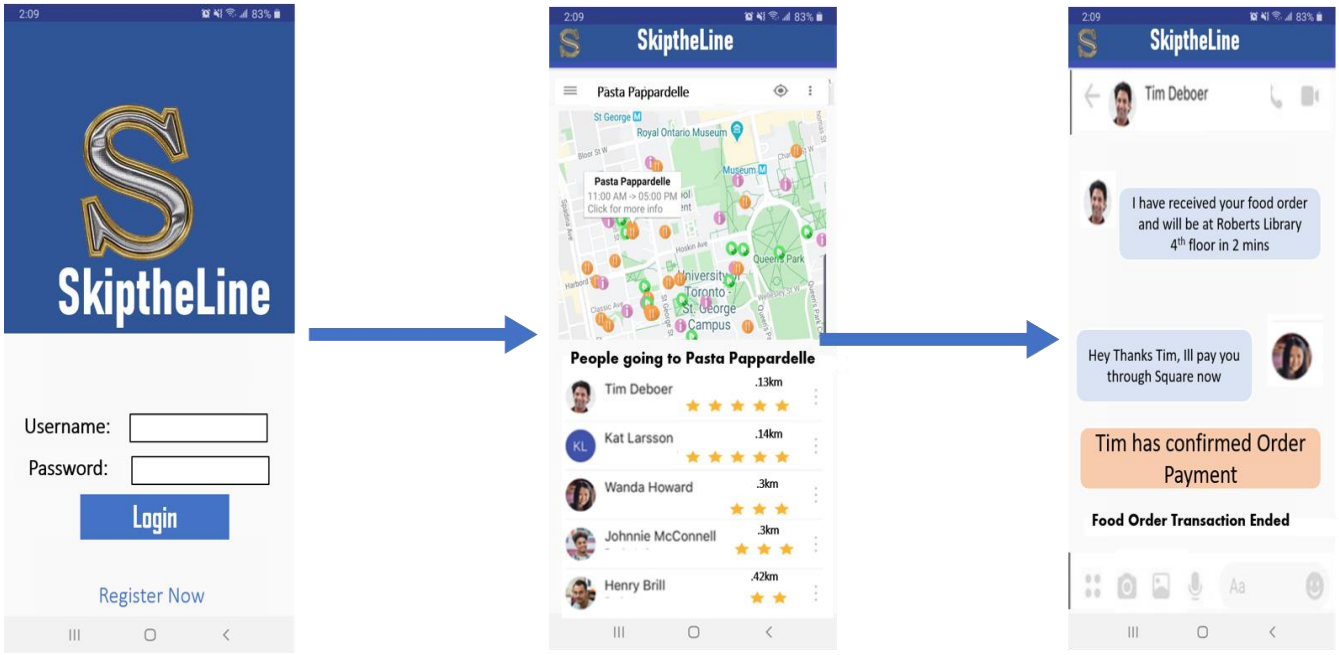
SkiptheLine App

Improve your food consumption experience at UofT



System Requirements and Architectural Design

Proof of Concept

Rafia Aqil, Celestina Azuure, Faria Khandaker, Zhaotian Li, Danni Zhou

Summer 2020

Contents

1 THE PURPOSE OF THE PROJECT

1.1 THE USER BUSINESS OR BACKGROUND OF THE PROJECT EFFORT

The St George Campus of the University of Toronto contains over 30 eateries for students to choose their meals from; each located at different parts of the campus. The eateries are managed by UofT Food Services. Students also have the option of purchasing food from many independent food trucks that are located around campus. Despite their popularity, UofT Food Services do not keep track of the food trucks' locations or menu items on their website. While there is variety in the options available, retrieving the food is a struggle for students.

The university tries to make food affordable and accessible to people with different dietary restrictions but because food outlets are often far away or have long lines, students struggle with the procurement of a meal. This results in students often having to leave their study spot in the library, asking strangers to look over their possessions, or missing the first few minutes of class because they were in line for a meal.
Eateries on campus do not support external food delivery apps like Skip the Dishes or DoorDash and this also a source of inconvenience for students

There are two applications currently available to UofT Students to help them in attaining a meal. The first is UofT Foods which allows students to search eateries by UofT campus and displays the hours and physical location of the selected eatery. The second is TMap (for UofT). This application is designed to be an app which displays the locations of buildings, eateries, bike racks and other facilities around different UofT Campuses on a Google Map like interface. Unfortunately, none of the apps allow the users to view the menus of any eatery and the information on food trucks is not available at all.

Our SkiptheLine App addresses the features that are lacking in the current applications by having a centralized system where students can view the locations of all the eateries and food trucks on campus, view the items on their menus, order food without leaving their study spaces and request food drop off by other app users via a messaging platform. It improves the food consumption experience on campus by decreasing the obstacles students face in their food procurement journey. This is achieved by facilitating live data collection in the form of orders and location to streamline and improve the process of ordering food on campus. This type of improvement can also increase the revenue that is brought in by food services as the convenience introduced by SkiptheLine can encourage more students to order food from campus.

1.2 GOALS OF THE PROJECT

There are two main goals we attempt to achieve through the SkiptheLine System:

1. To facilitate the process of procuring a meal or a snack and making it as convenient as possible

What is addressed: Students must no longer leave their belongings with strangers or miss class while standing in line

How to measure this goal: Graphing out App usage statistics and comparing app usage during exam season to non-exam season times. An increase in app usage around exam time can be correlated with people deciding to stay in their study spots and asking other students to take their orders via the App. Another measurement could be how many customers were not matched with a client around exam season. Perhaps less people are willing to move from their study spaces and therefore less clients available to take customer orders.

2. Have a centralized location for students to view menus of on-campus eateries and food trucks

What is addressed: Students may not be aware of all available options on campus and this can be useful for students who have dietary restrictions and may be able to add more variety to their diet

How to measure this goal: An increase in user base that doesn't correlate to app usage can indicate that some students are only using the application for the information it contains, not necessarily for the service it provides.

2 THE STAKEHOLDERS

2.1 THE CLIENT

Although the project is designed for the University of Toronto community, it is not commissioned by any entity and it is designed to solve a problem faced by the student body. Therefore it is difficult to define a "client": for now we, the start-up team, are the client (sponsors).

In our project, we use the term "clients" to refer to the app users who identify themselves as the voluntary delivery person.

2.2 THE CUSTOMER

Our customers are University of Toronto members, including enrolled students, faculty members, staff members and librarians.

In our project, we use the term "customers" to refer to the app users who place orders with the voluntary delivery person, i.e. "clients".

2.3 Other Stakeholders

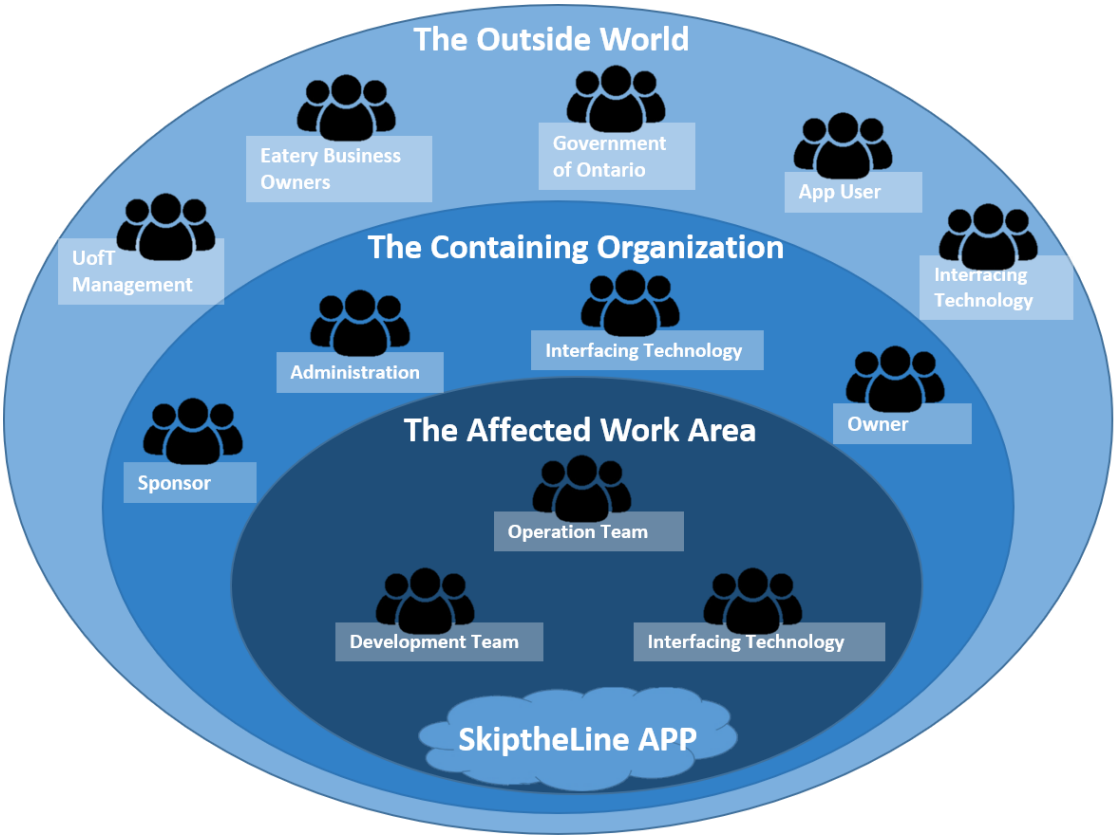


Figure 1: Onion Diagram of Stakeholders of SkiptheLine

#	Stakeholder Identification	Knowledge needed by the project	Degree of involvement	Degree of influence	Agreement on how to resolve conflicts with stakeholders with similar interests
1	Development team	Software development skills in frontend, backend, UXD, etc.	High	High	Work closely with to ensure they are in agreement and support change
2	Operation team	Marketing, sales, customer management, public relations, etc.	High	High	Work closely with to ensure they are in agreement and support change

3	Administration	Accounting, legal, human resource and other supportive functions when necessary	Medium	Medium	Inform and consult
4	Owner	Knowledge and vision of the product and the market	High	High	Work closely with to ensure they are in agreement and support change
5	Sponsor	Fund, experience, networking	Medium	High	Keep informed and satisfied
6	App user	E-mail (for registration and authentication), dining preference	Medium	Medium	Inform only when necessary; monitor and engage with limited range of topics
7	Eatery business owner	Eatery information and menu	Low	Low	Inform and monitor
8	U of T Management	Relevant regulations for on-campus business operation, special order and rules of public area (like libraries)	Low	Medium	Inform and consult when necessary, keep satisfied
9	Government of Ontario	Relevant regulations on food processing and delivery, such as hygiene and safety concerns	Low	Medium	Inform and consult when necessary, keep satisfied

3 CONSTRAINTS

3.1 SOLUTION CONSTRAINTS

Description	The system must be compatible with the latest version of iOS and Android operating systems including their previous versions.
Rationale	The operating system has the largest market share and currently represents the majority of mobile phone users. About 88% of the global market share constitute android users while 11.9% are iOS users (Gartner, 2019)
Fit Criteria	The system must comply with relevant standards of iOS and Android and support their user interface design guidelines to operate successfully.

Description	The system shall integrate with Google to facilitate real time location updates
Rationale	Google Maps is recently the most popular and preferred maps application used by the majority of smartphone owners (67%) for location tracking (Riley Panko, 2018).
Fit Criteria	The system must comply with the standards of Google and their interface design guidelines to operate successfully.

Description	The system must operate on desktop computers having a normal CPU clock speed of 3.5 GHz and a minimum of 16GB of RAM and storage of 128GB on a hard drive or SSD.
Rationale	The system will be developed and supported by the operations and development team. Therefore, it must fit into the specifications of the tools that will be used to develop and maintain the system, i.e. the development and maintenance should not require more RAM than 16GB.
Fit Criteria	The system's processing demands must be within an allowable range of the device's limits in capacity and storage space.

Description	The system shall integrate with my MySQL to ensure real-time processing of data
Rationale	It has the capacity to accurately manipulate, process, store and retrieve data within seconds
Fit Criteria	The system must accurately process and retrieve app user data in SkiptheLine within 2 seconds after inputting the data. The system interface must be designed to integrate with MySQL.

Description	The product shall facilitate automated processing of payment through smooth

	integration of the Square payment portal for app users who choose to make payment within the app
Rationale	The app user shall not be required to waste time navigating the payment process and shall complete the payment process easily at their own pace.
Fit Criteria	The system shall allow app users who choose to use the app payment system, make payments without leaving the app. The app shall allow conversion from foreign currency.

Description	The system shall allow the app user to submit messages within the SkiptheLine app, through a smooth integration of the Applozic instant messaging system.
Rationale	The app user will be able to submit messages or chats without being redirected to another communication medium or having to use telephone or email.
Fit Criteria	The user must be able to submit messages or chat within the app, without being redirected.

Description	The system shall integrate with CA Strong authentication to facilitate the validation of user credential and protection of user information
Rationale	CA Strong is one of the most reliable forms of multifactor authentication that provides maximum validation and protection of user information from breaches.
Fit Criteria	The system must authenticate all users of the app and shall allow users the option to select from a variety of authentication methods they want to. These include passwords, SMS, email, knowledge-based authentication, two-factor software tokens, and hardware credentials.

3.2 Implementation Environment of the Current System

Currently, there is no food ordering application at U of T campus that provides a one-stop-shop access to eateries and offers the opportunity for the U of T community to order food. The eateries on campus do not support external delivery apps and the current U of T foods application provides empty eateries information with no access to the smart food ordering process. SkiptheLine application will therefore be deployed on mobile applications and will require an effective internet to access a centralised location that facilitates a convenient food procurement process. All mobile app users will have SkiptheLine information displayed to them through the device that integrates with APIs that can be built using JSON, JavaScript and Spring MVC within the context of SkiptheLine application.

3.3 Partner or Collaborative Applications

The system will partner with Google Maps to facilitate accurate real time location tracking of customers during food delivery as well as the locating of eateries on U of T campus. Google Maps is widely used by the majority of smartphone users and has an effective and easy-to-use interface that can integrate well with SkiptheLine. SkiptheLine will also integrate with Square Payment solutions to facilitate a smooth and automatic processing of payment for SkiptheLine users who prefer to make payment within the app without redirecting them. In addition, the app will partner with CA Strong to provide a secured validation, authentication, and protection of users' information from unauthorised access, MySQL to ensure real time data manipulation and Applozic for instant messaging and feedback.

3.4 Off-the-Shelf Software

The owner will be adopting an off-the-shelf software for SkiptheLine system. And the system will implement the distributed messaging software to facilitate the processing of information inputted. The system shall also incorporate Cloud database services to store and access user and app information.

3.5 Anticipated Workplace Environment

SkiptheLine has incorporated all the relevant software components needed for it to effectively operate. Therefore, mobile users do not require additional hardware or software equipment to use the system. In this regard, a smart mobile device that meets the required operating systems highlighted section 3.1 and an internet connection to support the use of the features is enough.

For the operations team using the web-based version to manually update eatery information would be a desktop that meets the capacity of normal good CPU clock speed of 3.4 GHz and a minimum of 16GB of RAM and storage of 128GB on a hard drive or SSD. The SkiptheLine system is to be operated in an office where it is easily accessible by the SkiptheLine operations team. The user would have to be working in a position that accommodates for continuous work hours as their role of inputting food eatery information may be required all the time.

3.6 Budget Constraints

The owner of SkiptheLine has secured a funding of $250,000 towards the development of SkiptheLine application and including additional services. This funds however does not include some key vendors, with whom negotiations are in the process. The table below contains the estimated cost structure of the application within one year.

Vendors	Cost structure	Total estimated cost
SkiptheLine Team	None	$250,000
Google Maps	Free up to 40,000 calls for directions per month $5 for every 1,000 calls between 40,000 - 100,000 calls	$1200
Apache Kafka	~ $500 per month	$6,000
Integrated Square Payment	$3.4% + 0.15% per manually entered transaction	$0
Applozic	Pro plan $449/month	$5,388
MySQL	$2,000 per year price starting	$2,000
Total estimated Cost		**$264,588**
Cloud database	To be determined in negotiation with vendor	TBD
CA Strong Authentication	To be determined in negotiation with vendor	TBD

*Payment has no monthly fee, trans

3.7 SCHEDULE CONSTRAINTS

SkiptheLine shall be deployed on December 1, 2020 which is one month prior to the start of Winter term of 2020/2021 academic year (January 7, 2021). This period is adopted to allow SkiptheLine to be tested with a few actual users to enable potential glitches in the product to be detected and rectified by the beginning of the Winter School Term. The table below gives a summary of the time schedule.

Event	Deadline	Deadline Rational	Impact if deadline is unmet
Project Launch	July 30, 2020	Provides enough time to prototype, test, evaluate, and to launch the product by January 7, 2021.	Probable delays to the product launch date
SkiptheLine application ready for iOS/Android. Admin Module Completed.	August 30, 2020	Allow enough time for both mobile and admin modules to be	Possible delays to the final launch of the product.

		developed and made ready for official testing	
Mobile Application and Admin Module Testing	September 20, 2020	Time allocated for rigorous testing towards completing the beta version	Possible delays to the final product launch. It could affect app final performance
Beta Version Completed	October 20, 2020	Enough time completing the beta version and training, evaluation, and final deployment.	Potential delays to the final launch of the product. Beta version cannot inform final product deployment.
Training for SkiptheLine operations team and administrators	October 31, 2020	Provide sufficient training on product functionality	Inadequate training can affect easy usability. Can cause customer dissatisfaction and app performance.
Evaluation of Beta Version & rectification of issues	November 20, 2020	Enable version to be deployed with enough time to rectify issues prior to product launch.	Probable delays to product final launch. Inability to recognise issues and address them.
Final Product launch	December 1, 2020	Allow for the app to be tested at the Faculty of Information prior to the beginning of Winter term to correct issues before final launch, and to ensure that product is available to students during the Winter term.	Failure to perfect the product by the start of Winter term. Delays in familiarizing users with app No time to market products to the U of T community.

4 NAMING CONVENTIONS AND TERMINOLOGY

4.1 GLOSSARY OF ALL TERMS, INCLUDING ACRONYMS, USED BY STAKEHOLDERS INVOLVED IN THE PROJECT

Term	Definition
App user	The general term for those who register on SkiptheLine and may use the app as either customer and client, as defined below
Customer	App users who place order for food purchase and delivery service
Client	App users who declare that they are available and willing to provide food purchase and delivery service
Eatery	All the food providers geographically on or near campus that are frequently visited by U of T members, include but not limited to U of T dining halls/canteens, food trucks, independent local restaurants, chains like Starbucks and Tim Hortons, etc.;
Development team	People who build SkiptheLine as software engineers working on system architecture design, backend (database) and frontend, technical support, UX designer, etc.
Operation team	People who are responsible for the data maintenance and customer relations of SkiptheLine, such as collecting and updating eatery information, handling customer issues, coordinating with external stakeholders, marketing and public relations, etc.
Administration	Supportive functionalities of the business such as accounting, legal, human resource, etc. that can be outsourced to third parties.
API	An application programming interface (API) is a computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. (2020, Wikipedia)

5 RELEVANT FACTS AND ASSUMPTIONS

5.1 RELEVANT FACTS

According to the official data from the U of T website, as of 2020 there are 93,081 students, 14,648 faculty members, 7,405 staff members and 158 librarians. We use 50,000 as an estimation of our target MAU (approximately 50% of students plus 30% of faculty, staff members and librarians).

There are more than 30 eateries on St. George campus, and another 10-20 food trucks along St. George Street. The restaurants within walking distance are also frequently visited by U of T members. An estimation of 80-90 eateries might be shown on the radar of SkiptheLine.

5.2 BUSINESS RULES

- Primary users shall be U of T members, and the operation area is limited to on-campus locations (meaning that you cannot order food to your off-campus residence on SkiptheLine);
- By using SkiptheLine service, clients are willing to help others when it is convenient on their way to and back from the eatery, not intentionally accepting orders and expecting tips from customers;
- Customers shall clearly specify the food item and any requirements when placing the order, and the app will save records of the order detail in case of unsatisfied experience dispute;
- Customers and clients both have the right to choose or deny delivery service before an order is officially placed;
- Customers and clients both shall respond to each other's message promptly when it is possible and in a safe circumstance;
- Clients shall order and bring the food item according to the order detail, bring the receipt together with the order to show proof of payment;
- By using SkiptheLine service, customers acknowledge that they are willing to pay tips to show their gratitude, but they are not forced to pay a certain amount of money for each order;
- Customers shall rate the client on a fair and reasonable basis;
- The review system is designed to prevent dishonesty issues from happening. Any abusive misuse or personal attack shall be avoided by methods like user education (app usage principle/agreement) and manual intervention.

5.3 ASSUMPTIONS

- In the U of T community, people have a sense of belonging and tend to trust each other in most cases; students, faculty and staff members are good people who want to help each other out;
- U of T members that show up on campus shall be healthy and do not carry infectious disease that spreads via air, touch and other ways involved in food delivery process (i.e. people with severe infectious illness shall not come to school in the first place);
- Unlike Uber Eats and Doordash, app users are not joining this community for the purpose of making money, they are not doing this to make a livelihood, just to collect some pocket change and help others when it's possible;
- Eatery owners or management are willing to share menus with SkiptheLine users, as the menus are visible to everyone online and at the eateries themselves, and by sharing the menu SkiptheLine may even bring extra sales to the eateries;
- U of T management agrees that SkiptheLine can operate on campus to serve the community, under the condition that all users obey the rules in public areas and do not disturb the daily operations of the university;
- While SkiptheLine operation team will do its best to monitor unwanted accidents of privacy and security issues, SkiptheLine is only a platform or community that brings together those in need and those who can help, and it cannot be responsible for certain types of accidents that are out of its control;
- The whole community and stakeholders hold a positive view of SkiptheLine and are willing to make contributions.

6 THE SCOPE OF THE WORK

6.1 THE CURRENT SITUATION

In the present situation, an easily accessible way for students to find out about the menu items of eateries on campus does not exist. Unless a student is aware that they can access the menus of eateries from UofT Food Services Website, they will have difficulty locating menus for all eateries on campus. The Food services website does not contain the menu of food trucks. There is not a simple way to compare eatery items and prices. Students do not have a way of thinking about their food choices prior to entering an eatery nor do they have the option to mobile order their food to avoid standing in line and missing class time. There are two applications available to UofT students for their mobile phones which can provide basic information like the address of the eatery and its hours of operation but none of the apps have information on food items or their prices.

In order to address the issues in the current situation, the SkiptheLine team is required to develop an app that allows its users to view all the menu items of all on-campus eateries/food trucks, view the distance of the eateries/food trucks from the user's current location, track app users who are willing to accept other app user's food orders (client) and deliver the items safely to the second app user (customer). The app will also contain a messaging platform that allows the client and customer to chat with each other about specific order details and finally a rating system which allows customers and clients to rate each other's customer service and professionalism.

6.2 THE CONTEXT OF THE WORK

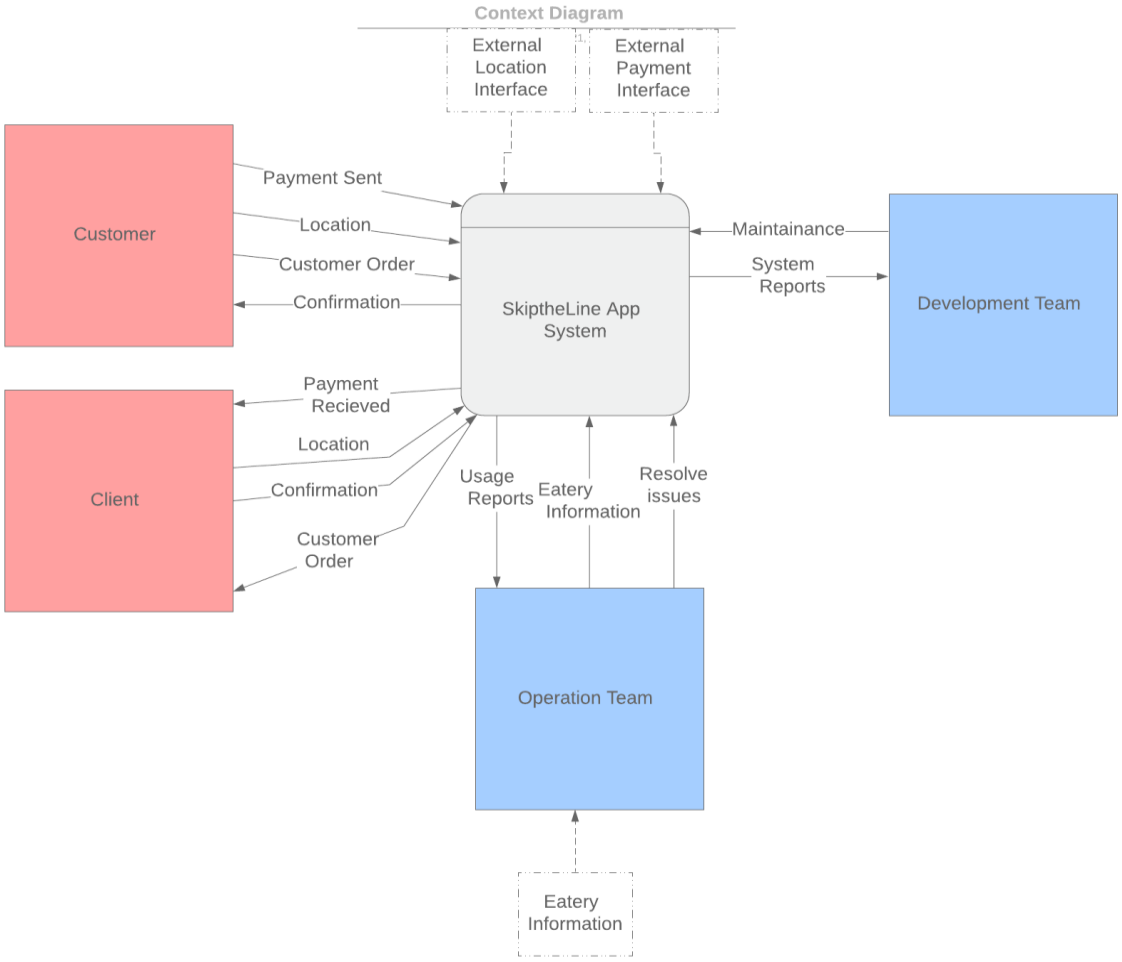


Figure 2: Context Diagram of SkiptheLine System

The App (Figure 2) is the centralized system that takes care of all information delivery about food services on campus. There are four parties who directly interact with the app: the customer, the client, the operations team whose job is to manage administrative tasks like keeping menu items and prices up to date and the development team whose job is to keep the app running smoothly.

The app facilitates communication between the client and the customer. Both the Client and the Customer are users of the app, customers are the ones who order food, clients are the ones who deliver food. Customers tell clients what they want to order via a messaging feature on the app. Clients inform customers when they have retrieved the food order and are ready for the exchange of food items for money, also using the messaging feature. Although app users have the option to pay for the food items using the inbuilt payment system, users also have the flexibility to use Peer to Peer payment options such as UofT's T-Bucks, hard cash or e-transfer, at their own discretion. The app serves to facilitate communication between app users. Communication with external parties like the eateries are done via the app users; the app itself has no direct link to food service providers.

6.3 WORK PARTITIONING

No.	Event Name	Input	Output	BUC Summary
1	App user creates a profile	- Download the SkiptheLine app and register.	- Registration confirmation	App user creates/updates profile information and is ready to use SkiptheLine services.
2	Customer searches for an eatery and a Client	- Current time and location	- Eatery and menu info - Available client(s) - Each client's rating	Customers find the open eateries and available clients nearby and choose the client.
3	Customer Places Order and Client receives order	- Food order information	- Order Confirmation	Customer finds Client and places order and Client confirms.
4	Client searches for Customer	- Client purchase confirmation - Customer's location	- Delivery confirmation	Client locates customers and meets them in person.
5	Customer makes payment and Client receives payment	- Receipt from eatery	- Payment	Client hands over customer order and receipt from eatery to customer. Customer proceeds to make a payment using the available payment methods.

7 BUSINESS DATA MODEL AND DATA DICTIONARY

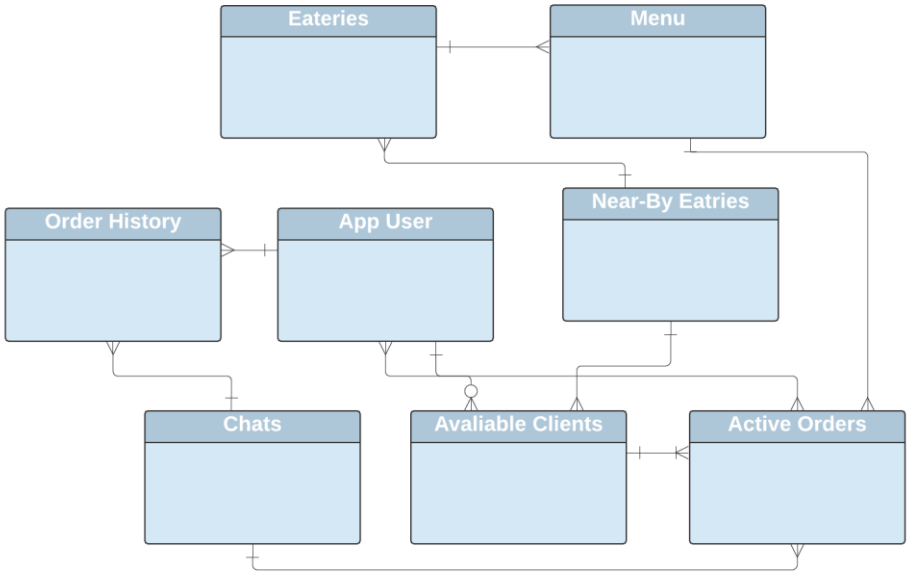


Figure 3: Business Data Model for SkiptheLine Mobile App

The Business Data Model illustrates all relevant tables required for SkiptheLine domain. The following table provides a detailed data dictionary for the tables illustrated in figure 3.

Data Entities/Attributes	Description
App User	**All App User information. Also used to validate App User upon login**
AppUserID	App User's unique ID (used by Administration Team)
AU_Username	App User's unique username for profile
AU_Password	App User's password (used for logging into account)
AU_Email	App User's email
AU-PhoneNumber	App User's phone number
AU_ProfilePicture	App User's picture
AU_Rating	App User's rating (by other App Users)
AU_Client_Status	Boolean data-type to classify App User as Client
AU_Location	Stores App User's current location

Available Clients	**All App Users information that identified as Client**
AvaliableClientID	Unique identifier for Available Clients, in order to distinguish clients other App Users
AppUserID	Foreign key identifier; carries all 'App User' table attributes
EateriesID as 'Going to'	Foreign key identifier; carries all Eateries table attributes
NearByEateriesID	Foreign key identifier; carries all 'Near By Eateries 'attributes
AC_CurrentLocation	Retrieves current location of Client
AC_DistanceEatery	Computes distance between eatery and client
AC_DistanceCustomer	Computes distance between customer (App User that is viewing Nearby Clients Page) and client
Active Orders	**All active order information**
ActiveOrdersID	Unique identifier for active orders
AvaliableClientID as 'Client'	Foreign key from 'Available Clients', gives the client an ID and helps distinguish client ID from customer ID.
AppUserID as "Customer"	Precondition: Customer initiating Order, their ID will be saved in this column. {In order to distinguish ID from Client and Customer}
EateriesID	Foreign key identifier; carries all 'Eateries' table attributes
MenuID	Foreign key identifier; carries all 'Menu' table attributes
AO_Date	Returns Date as Integer (YYYY/MM/DD), provides date when order was initiated
AO_ Initiate	Returns Time as String (hh:mm:ss), provides time when order was initiated
AO_Delivered	Returns Time as String (hh:mm:ss), provides time when order was delivered
AO_ClientStatus	Drop down list for client to update their status: Finding Customer, Ordered Food, Walking to Eatery
AO_Paid	Boolean data type. Returns True when customer pays client for food order & tip and client confirms payment reviewed.
ChatID	Foreign key identifier; carries all 'Chat' table attributes
AO_Total	Money data type. Holds total amount customer needs to pay client
AO_Payment_Method	Drop down list for the method used to pay: tBucks, e-transfer, Square
Chats	**Saved chat between Customer and Client**

ChatID	Unique Identifier for chats
Chat_Messages	Instant Massager's saved messages between customer and client regarding food order
Order History	**Saved order information transferred from 'Active Orders' table**
OrderHistoryID	Unique identifier for order history
ActiveOrdersID	Foreign key Identifier; carries all 'Active Orders' table attributes and stores into' Order History' table
Near-By Eateries	**All eateries located near App User**
NearByEateriesID	Unique identifier for 'Near-By Eateries' table
AvaliableClientID	Foreign key identifier; carries all 'Available Clients' table attributes
EateriesID	Foreign key Identifier; carries all 'Eateries' table attributes
NBE_Distance	Computes distance between App User and eateries
Eateries	**Information for all Eatery Restaurants located around the 3 campuses**
EateriesID	Unique Identifier for all eatery Restaurants
NearByEateriesID	Foreign key Identifier; carries all 'Near By Eateries' table attributes
E_Name	Eatery restaurant's name
E_Location	Eatery restaurant's location
E_Hours	Eatery restaurant's hour of operation
Menu	**Provides information on all restaurant's menu items for food and beverages**
MenuID	Unique identifier for all menu items
EateriesID	Foreign key identifier; carries all 'Eateries' table attributes
M_Iteamname	Name of the food items
M_Price	Money data type. Price of the food item.
M_Size	Size of the food item

8 THE SCOPE OF THE PRODUCT

8.1 STRATEGIC DEPENDENCIES

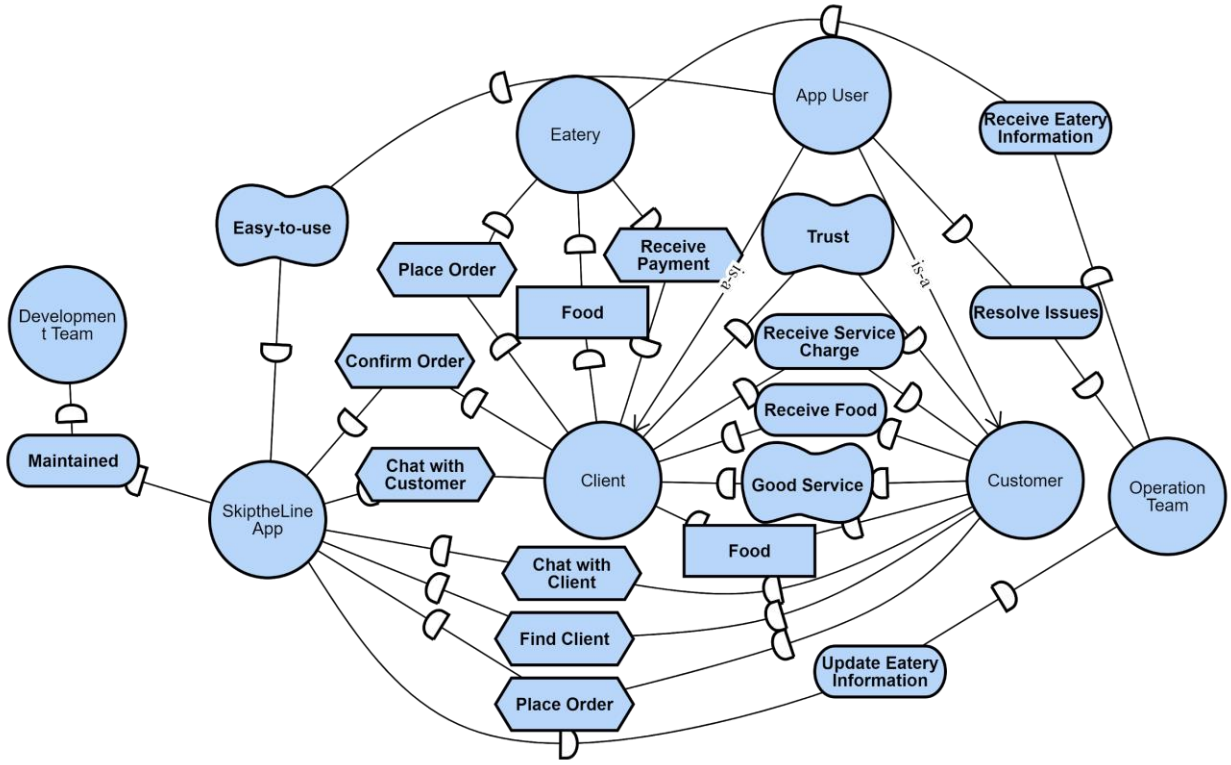


Figure 4: Strategic Dependency Model of SkiptheLine System Actors

The Strategic Dependency model illustrates the dependent relationship between the SkiptheLine app, App User (Client and Customer), Operations Team, Development Team and Eatery. The Customer and the Client depend on SkiptheLine app to place an order, communicate with each other, and make a payment. They rely on the Operations Team to resolve issues during this process. The Development Team is responsible for the maintenance of the app.

8.2 PRODUCT BOUNDARY

8.2.1 Use Case Diagram

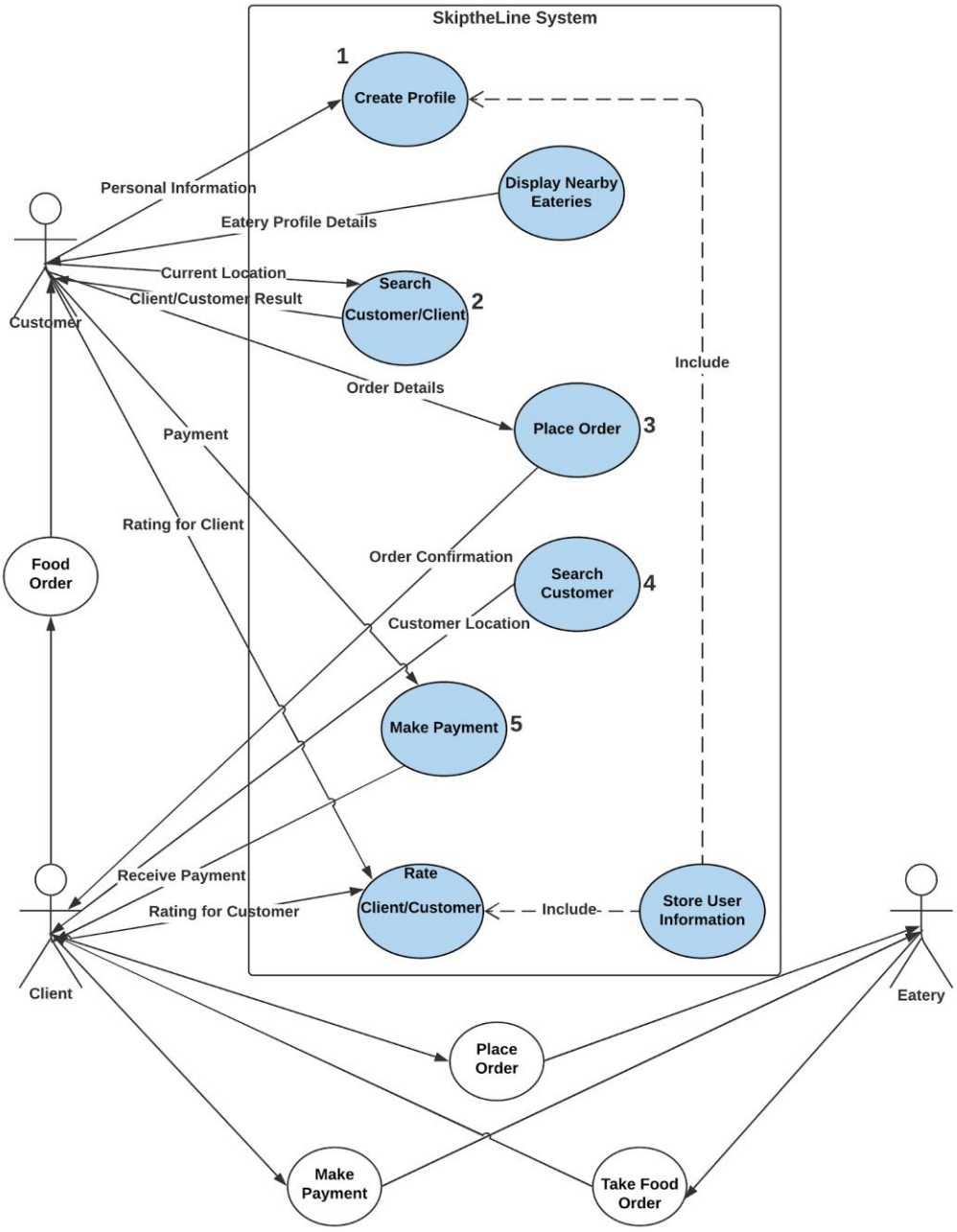


Figure 5: Use Case Diagram of SkiptheLine System

The Use Case diagram represents the primary users' (Customer and Client) interactions with the SkiptheLine system to complete an order. The eateries have no direct interaction with the SkiptheLine system. All necessary interactions with the eatery happen via the client. Section 8.4 below will illustrate the 5 main product use cases in this diagram: Create profile, Search for Client, Place order, Search for Customer, Make payment.

8.2.2 Product Scope Diagram

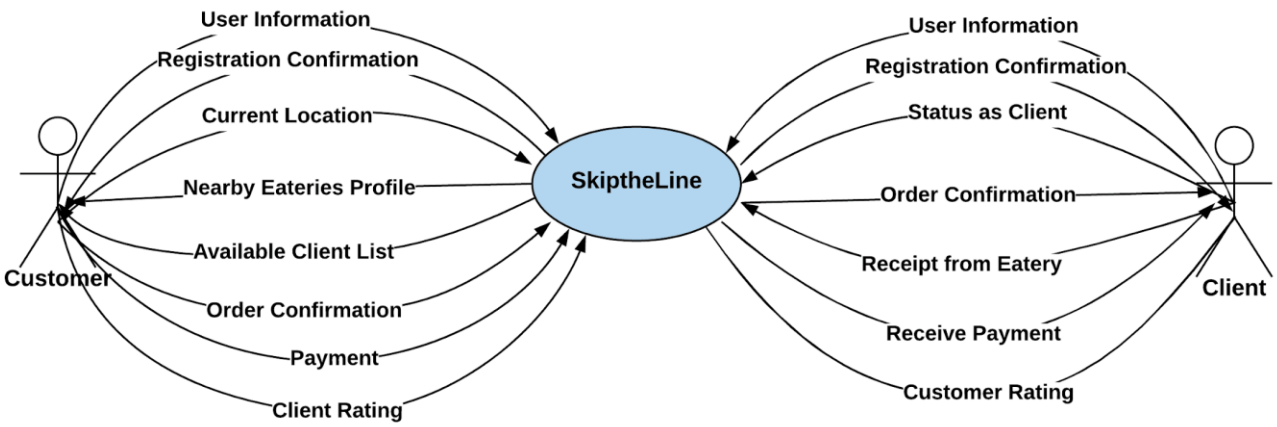


Figure 6: Product Scope Diagram of SkiptheLine System

The Product Scope diagram summarises the interfaces between SkiptheLine and the Customer and Client. It illustrates the workflow and data exchange between the users and the SkiptheLine system.

8.2.3 Page Navigation Design

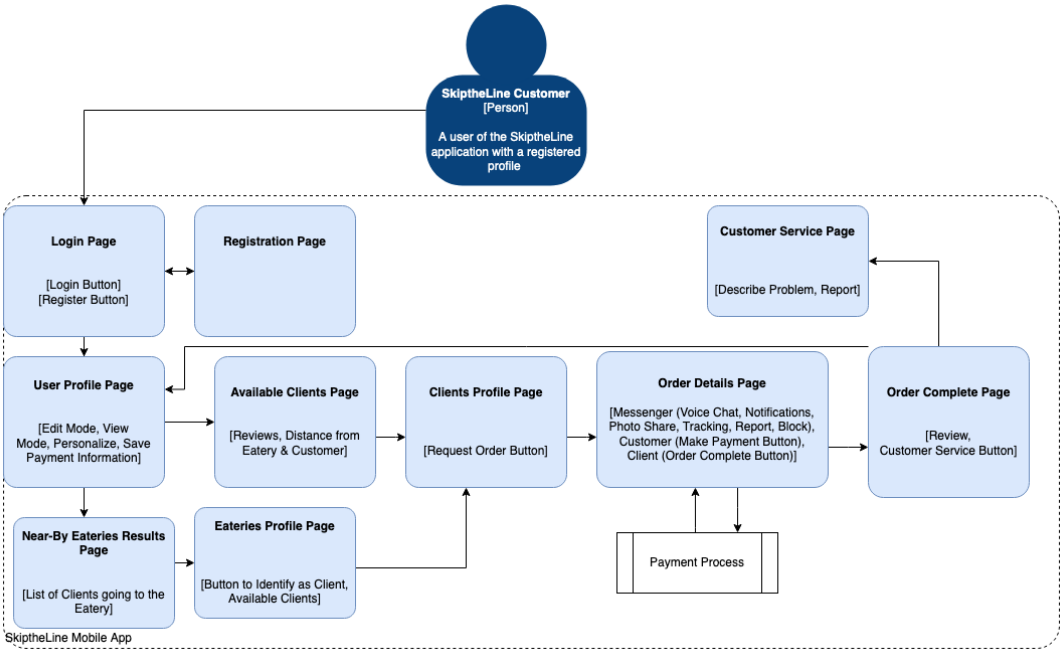


Figure 7: Page Navigation Design for SkiptheLine Mobile App

Figure 7 illustrates the page navigation design process from the start page to the end for the App User's.

8.3 Product Use Case Table

PUC #	PUC Name	Actors/Users	Input/Output
1	Create Profile	App User	User Information (in) Registration Confirmation (out)
2	Search for Client	Customer Client	Current location (in) Eatery Profile (out) Available Client list (out)
3	Place Order	Customer Client	Order Information (in) Order Confirmation (out)
4	Search for Customer	Client Customer	Customer's location (in) Delivery Confirmation (out)
5	Make Payment	Customer Client	Receipt from Eatery (in) Payment Confirmation (out)

8.4 PRODUCT USE CASES

Product Use Case: Create Profile

Section	Comment
Use Case Name	Create a SkiptheLine profile
Scope	App user interface: App user inputs their personal information, username, and password, verifies and registers in order to use the services in the SkiptheLine application
Level	App users aim to have their information saved securely and be able to access SkiptheLine services whenever they want.
Primary Actor	App user (Customer or Client)
Stakeholders and Interests	**App user:** the app user wants to create a SkiptheLine profile account in order to quickly find eateries and order for food. They wish to have their information saved for future use and stay up to date on the application related developments. **SkiptheLine and owner:** the actors want for the user to have a smooth profile account creation experience to ensure that they use their service efficiently
Preconditions	1. The app user has installed the SkiptheLine application on their device. 2. The app user has an effective internet connection.

Success Guarantee	1. The SkiptheLine user profile is created and all user details are saved to the database. 2. The user is a registered entity within SkiptheLine application and successfully logs in to their profile.
Main Success Scenario	1. The **app user** opens the app on their Android or Apple mobile phone device, 2. The **SkiptheLine system** opens the welcome page and prompts the user to 'Sign In' or 'Register Profile' as a. Customer b. Client 3. The **app user** selects the 'Register Profile' option, 4. The SkiptheLine **system** prompts the **app user** to: ● enter first and last name (required), ● enter their valid email address (required), ● enter a unique username (required), ● enter a password between using 8 or more characters with a mix of letters, numbers, and symbols (required), ● re-enter the password (required), ● enter their valid phone number (required), ● enter their valid payment information **(optional),** ● upload recent photo (**optional**: gives priority and adds credibility and services) ● swipe your finger print (**optional**) ● check the checkbox that confirms the user has read and agrees to SkiptheLine 'Terms of Use' and 'Privacy Policy'. 5. The **app user** enters this information and selects 'Register', 6. The SkiptheLine **system** validates that all required information has been provided correctly, and the username is unique within the existing database, 7. The **SkiptheLine system** creates the user profile, assigns it a unique ID, and stores the profile information securely within the database, 8. The **SkiptheLine system** prompts the user to swipe their finger print and enter username and password. or the system sends a verification email to the user and displays a notice: 'Thank you for joining **SkiptheLine**! To complete your registration, please verify your profile within the next 48 hours by clicking on the link that we have sent to your email [email provided by user]. If you have not received an email from us, please check your Junk/Spam folder', 9. The **app user** receives the confirmation email, backs out of the **SkiptheLine** application, and verifies their profile by clicking on the link provided, 10. The **SkiptheLine system** opens a new webpage on the user's phone via the link and displays a notice: 'Thank you, your profile has successfully been registered. Please return to the **SkiptheLine** app and complete your login using your account information',

	11. The **app user** backs out of their email and opens the application, 12. The **app user** selects the 'Sign In' option, enters their username and password, and successfully logs in, 13. The SkiptheLine **system** displays the application homepage.
Extensions	6a. The app user does not enter the required information in a valid format or fails to check the 'Terms of Use' and 'Privacy Policy' checkbox. ● An error message is displayed on the page: 'Please complete all required fields to proceed with the registration of your profile. Please verify that this information has been entered in the correct format'. ● The app user is directed to the area where the first instance of required information is missing. 6b. The app user enters an already registered unique username. ● An error message is displayed on the page: 'This username is already registered with us. Please try another username or sign in'. ● The user is directed to the area where the email information is located for re-entry. 6c. The app user inputs an already registered email address that exists in the system. ● The system prompts the user that the "Email address already exists". ● The system directs user to enter a different email address. ● The system directs app user to sign in or register. 6d. The app user signs in with an unregistered username and password ● An error message is displayed on the page: This username and password is not registered with us. Please fill in all required information and register with us ● User is redirected to fill in all required information and register 6e. The app user uploads someone else's photo ● The SkiptheLine system prompts user to retake photo directly on the app and reupload ● The SkiptheLine System blocks user from seeing the previous photo 7a. The system fails to create a user profile, despite the required information being successfully provided by the user. ● A report including the system log will be sent for review by system administrators. ● An error message is displayed on the page: 'Unfortunately, we can not create your profile at this time. Please try again later'. 9a. The app user fails to verify their account within 48 hours. ● The system removes the profile from the database.

	● The link is modified to display an error message: 'Oops! It looks like you were not able to verify your account in time. Please complete the registration process again to create a profile with us. Thank you!' 12a. The app user provides incorrect login information 7 consecutive times. ● The system locks the account and directs the user to reset their password.
Special Requirements	1. The SkiptheLine system must store the user information securely. 2. The system must be accessible to people with vision impairments.
Miscellaneous	An existing app user needs to just enter their username and password to sign in anytime they want to use the app.
Autor	App user (customer and client), owner

Product Use Case: Search for Client

Section	Comment
Use Case Name	Client and Customer searches for a matching Client
Scope	Client updates status and Customer searches for a matching Client for food ordering and delivery
Level	Customer can get the food without leaving where he/she stays (specification template)
Primary Actor	Client and Customer
Stakeholders and Interests	1) Development team: how to design the functions so that it's easy to use for both the Customer and Client; 2) Operation team: how to teach the users and promote the app, and how to improve the usability; 3) App users: how to use the app carefreely
Preconditions	Both the Client and Customer have access to their smartphones and access to the internet; both are registered users of SkiptheLine and are logged in
Success Guarantee	1) At least one person near the Customer plans to go to an eatery nearby and 2) is willing to help others and 3) updates his/her status on the app in time successfully (to be identified as "Client") and 4) the status and this Client's name and rating are seen by the Customer
Main Success Scenario	Customer finds the available Client(s), chooses him/her (or one of them) and decides to place an order with him/her.
Extensions	Customers cannot find anyone that is available for this request.

Special Requirements	The app suggests currently available Client(s) that plan(s) to go to other eateries.
Miscellaneous	The app can also recommend the popular menu items from the alternative eateries.
Author	Customer

Product Use Case: Place Order

Section	Comment
Use Case Name	Place Food Order with Client
Scope	Customer provides order details and waits for confirmation from Client through SkiptheLine messenger
Level	Customer can place food order through SkiptheLine App without physically going to the restaurant
Primary Actor	Customer and Client
Stakeholders and Interests	1) App Users (Customer, Client): Customer want to receive food order, Client want to fulfill Customer order & receive their service charge, 2) Eatery Business Owners: want revenue from food orders, 3) Development Team: want the interface be easy-to-use & app maintained, 4) Operation Team: want all eatery information updated & resolve app users issues.
Preconditions	1) App Users (Client, Customer) have access to the internet, downloaded SkiptheLine app and have registered accounts. 2) Client(s) have updated status.
Success Guarantee	1) Eatery information is up-to-date(discontinuations, change in menu, price changes). 2) Customer finds Client(s) going to their desired eatery. 3) Trust relationship between Client & Customer
Main Success Scenario	Client confirms taking Customer order
Extensions	1) If Client denies Customer order, then Customer find other Clients 2) If Customer messages several Client(s) and receives no response, then Customer physically goes to Eatery Restaurant for Food Order (and can become a Client)
Special Requirements	SkiptheLine messenger interface is easy to navigate and provides instant notifications for messages (help reduce wait time between Customer & Client)

Miscellaneous	Client is at the Eatery Restaurant and Customer's requested food order is unavailable
Author	Customer

Product Use Case: Search for Customer

Section	Comment
Use Case Name	Looking for the Customer
Scope	Client locates Customer and they meet in person
Level	To meet the Customer in person and deliver their order
Primary Actor	Client
Stakeholders and Interests	Client – to find Customer to deliver their food and receive service fee from them; Customer – to receive their order from Client; Operation Team – to resolve issues that may arise
Preconditions	The Client must have purchased the food Customer ordered
Success Guarantee	The Client is able to locate Customer; The Customer must be available to receive the order
Main Success Scenario	The Client finds the approximate location of Customer using the tracking feature of SkiptheLine and heads towards the location. The Client uses the Customer's photo to identify them (if they have one on their profile). And/or The Client communicates with the Customer using the chat feature of SkiptheLine to get detailed information that may help identifying the Customer (Customer's attire, surroundings,etc.) The Client is able to find the Customer and the Customer is available to receive the order at the moment.
Extensions	1. The Customer does not have a profile photo and is not being responsive when the Client tries to communicate. The Client gives the Customer a lower rating. 2. When the Client arrives, the Customer is still in a lecture/meeting/appointment and is not available to accept the order. The Client leaves the order at a safe place which both parties have agreed upon. 3. The Client or the Customer is not cooperative but uses abusive languages when

	communicating. The Client reports to the system administrator and the Customer's account will be suspended upon investigation or vice versa.
Special Requirements	The App allows the Client to track the location of the Customer and vice versa.
Miscellaneous	Both Client and Customer must have access to internet connection to use the tracking and chat features of SkiptheLine.
Autor	Client

Product Use Case: Make Payment

Section	Comment
Use Case Name	Payment for Food Items
Scope	SkiptheLine App - Method of Payment for Goods
Level	Customer pays Client for delivered food and Client receive appropriate reimbursement
Primary Actor	Customer
Stakeholders and Interests	Customer wants to make payment for delivered food. Client wants to receive payment for delivered food
Preconditions	Mobile Phone with SkiptheLine App downloaded, connection to the internet, location enabled
Success Guarantee	An exchange of finances and food items occur in some form between Client and Customer
Main Success Scenario	Client and Customer meet, food items and receipt from the eatery is handed to Customer from Client. Client and Customer engage on an agreed upon method of financial transaction. Customer walks away with food, Client walks away with funds

Extensions	1. Customer selected pay-in-cash option on app but doesn't have enough cash once the client arrives a. Customer pays with credit card using the built-in payment system b. guest-deposits t-bucks into client t-card c. Interac-e-transfers money to client bank account 2. Customer decides to cancel order after client has purchased food a. Customer account receives bad rating b. Customer account is financially penalized 3. Client loses physical receipt from eatery a. Customer checks menu price from app and gives the amount specified on the menu and adds on tax b. Client took a photo of the receipt and uploaded it to the app cloud, showed the customer the picture and the customer paid the amount shown. 4. Customer requests payment method change to a form of payment that the client does not accept but client as already purchased food items a. client refuses to accept the change and customer must find a way to play client through the client's preferred method b. customer can cancel order and risk financial penalty on account
Special Requirements	1. Client and customer agree on method of payment, various payment options available 2. Client is headed to the same eatery that the customer wants the food from or is willing to head towards the desired eatery, even if it is not the client's original destination
Miscellaneous	1. Variation in food prices between app menu and menu at eatery 2. Customer assumed app menu included sales tax
Author	App Owner, App Users (Client and Customer)

9 FUNCTIONAL REQUIREMENTS

Requirement ID: *CAA01*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile

Description: The system must have the user profile page for users to create profiles with options to update profiles and save payment information. It shall ensure that each username is unique and does not already exist in the database.	
Rationale: To ensure users can register and each user within the SkiptheLine database can uniquely be identified, allowing proper user authentication and overall application functionality including security and integrity. This shall also ensure that all actions undertaken by a user can be traced back to an exclusive ID.	
Originator: System Administrators	
Fit Criterion: There shall never be a duplicate usernames within the SkiptheLine database at all times. The system shall prevent attempts to create a profile with a non-unique username.	
Supporting Materials: BUC #1, PUC Table (A2 Vision)	**Conflicts:TBD**

Requirement ID: *CAA02*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure secured authentication and validation of user credentials through email or multi-factor authentication or email		
Rationale: To ensure that there is a secured validation and authentication of user information within the SkiptheLine database.		
Originator: System Administrators		
Fit Criterion: The system shall require new users to swipe their fingerprint three times and must capture and store the fingerprint automatically in SkiptheLine system within 2 seconds after the last swipe. Each new user shall receive a verification email message and the system shall automatically generate the date and time of when the email was released. The system shall give the user the option to choose either biometric or email verification.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).		**Conflicts:TBD**

Requirement ID: *CAA03*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile
Description: The system shall terminate the validity of the verification link 48 hours following its release and delete the profile data connected to that user.		
Rationale: To ensure the efficiency of data storage and integrity of data within the SkiptheLine database.		
Originator: System Administrators		
Fit Criterion: No user profile account shall be verified over 48 hours after the verification email related to that profile was sent. Data related to unverified profiles shall be erased from the system database within 60 minutes following the 48-hour period.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).		**Conflicts:TBD**

Requirement ID: *CAA04*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall not allow for a user's username to be modified.		
Rationale: To ensure that each user within the SkiptheLine database can be uniquely identified, allowing appropriate user authentication and overall application functionality, and ensuring security. This shall also ensure that every activity carried out by a user can be traced back and linked to an exclusive ID.		
Originator: System Administrators		
Fit Criterion: There shall be no functionality that facilitates the modification of a username.		
Supporting Materials: BUC #1, PUC Table. (A2 Vision)	**Conflicts:TBD**	

Requirement ID: *CAA05*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall generate a report including the system log should a user be unable to create a profile despite providing all the proper requirements.		
Rationale: To ensure that issues with the profile creation process are raised and responded to quickly and appropriately.		
Originator: System Administrators		
Fit Criterion: A report shall be sent within 2 second to the System Administrators highlighting the error.		
Supporting Materials: BUC #1, PUC Table (A2 Vision)	**Conflicts:TBD**	

Requirement ID: *CAA06*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall display the login and ensure that all required information is provided by the user, and is in the correct format, prior to profile creation being complete.		
Rationale: To ensure the integrity of data within the SkiptheLine database and that users can be identified and contacted through the information provided.		
Originator: System Administrators		
Fit Criterion: Users shall receive a notification during the profile creation process should required information be missing or cataloged in an incorrect format. There shall be no missing or incorrectly formatted data within the SkiptheLine database.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).	**Conflicts:TBD**	

Requirement ID: *CAA07*	**Requirement Type**: Functional	**PUC:** Create a SkiptheLine profile account

Description: The system shall establish a secure connection to the SkiptheLine database to transfer personal data provided by the user during profile creation	
Rationale: To ensure that the security and privacy of users' personal information of users are protected against unauthorised access.	
Originator: System Administrators and Owner	
Fit Criterion: Users shall be required to enter their username and password to login and login shall not occur without these credentials. Information shall not be subject to a breach or hack.	
Supporting Materials: BUC #1, PUC Table, Onion diagram (A2 Vision).	**Conflicts:TBD**

Requirement ID: *CAA08*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall allow users to reset their profile passwords as needed.		
Rationale: To ensure that users have safe and private access to their SkiptheLine profile, and that personal information is secured		
Originator: System Administrators		
Fit Criterion: Users shall be presented with the option to reset their passwords via the 'Sign In' option on the welcome page. New passwords shall appear in the SkiptheLine database in 4 seconds or less.		
Supporting Materials: BUC #1, PUC Table (A2 Vision)		**Conflicts:TBD**

Requirement ID: *CAA09*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall allow users to change the email address associated with their username and profile as needed.		
Rationale: To ensure that users may continue using their existing SkiptheLine account and be contacted, irrespective of whether they change their email addresses.		
Originator: System Administrators		
Fit Criterion: Users shall be presented with the option to alter the email address associated with their profile. New email addresses shall appear in the SkiptheLine database in 4 seconds or less.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).		**Conflicts:TBD**

Requirement ID: *CAA10*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile account
Description: The system shall establish a secure connection to the SkiptheLine database to transfer personal data provided by the user during profile creation.		

Rationale: To ensure the privacy of user personal information and other sensitive information like payment options.	
Originator: System Administrators	
Fit Criterion: Information shall be transferred securely to the SkiptheLine database. Information shall not be subject to a breach or hack.	
Supporting Materials: BUC #1, PUC Table, Onion Diagram, SD Model (A2 Vision).	**Conflicts:TBD**

Requirement ID: ZL01	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall allow users (customer and client) to view the eateries nearby that are open and available.		
Rationale: Need to let users know the available options so that they can make decisions on what to eat.		
Originator: Customer and client		
Fit Criterion: When a user initiates a search, the app shall: ● provide a full list of eateries that are open on campus, based on current time and location (will be referred to as "search result list page"); ● by default, the list shall be sorted ascendingly according to the distance between each eatery and the customer's location.		
Supporting Materials: A2 Vision	**Conflicts:** TBD	

Requirement ID: ZL02	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall allow app users (customer and client) to select eateries based on their personal preferences.		
Rationale: To give users better knowledge of their options in terms of concerned aspects, so that they can make more informative decisions.		
Originator: Customer and client		
Fit Criterion: The search result list page shall be able to be sorted according to "number of available client(s)", "average price", "popularity/rating" or other features.		
Supporting Materials: A2 Vision	**Conflicts:** TBD	

Requirement ID: ZL03	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall allow app users (customer and client) to know further details about each eatery.		
Rationale: To give users better knowledge of the food options and client options, so that they can make final decisions on the order.		

Originator: Customer and client	
Fit Criterion: ● On the search result list page, users shall be able to click each eatery of the list to enter its information page; ● The information page shall contain details like location, open hours, available client list, menu items and prices, etc.; ● The information page shall have a button to go back to the list.	
Supporting Materials: A2 Vision	**Conflicts:** TBD

Requirement ID: ZL04	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall provide customers with available clients' detailed information.		
Rationale: To let customers know each client better for final decision.		
Originator: Customer		
Fit Criterion: On the eatery's information page: ● the list of available clients is sorted descending according to rating; ● customers shall be able to click the avatar of each client to view details like name, school/department, rating, etc. on his/her profile page; ● the page shall have a button to go back to the eatery's information page; ● a button to start chatting with this client, and; ● a button to place an order with him/her.		
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL20 in terms of privacy issue.	

Requirement ID: ZL05	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall allow app users to be identified as clients.		
Rationale: To encourage app users, and even those who originally want to use the app as customers, to become clients and help others.		
Originator: Client		
Fit Criterion: On the eatery's information page, there shall be a button in the available client list to let the app user sign up as a client for this eatery.		
Supporting Materials: A2 Vision	**Conflicts:** TBD	

Requirement ID: ZL06	**Requirement Type:** Functional	**PUC:** Search client
Description: In case of no client available for a certain eatery, the app shall suggest customers to wait for someone to come or become a client himself/herself.		

Rationale: To help customers solve the problem actively and try to let them stay with the app and open it whenever it's possible, since data like retention rate and DAU, user experience and word-of-mouth are all important indicators of success.	
Originator: Operation team	
Fit Criterion: If there is no client available for a certain eatery: ● there shall be a button on the eatery's information page that allows customers to receive notification for new client(s) going to this eatery; ● OR the app shall suggest the current user to sign up as a client as the other option.	
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL07 in terms of UX design (these 2 solutions are for the same problem, how to present them to the user without confusion?).

Requirement ID: ZL07	**Requirement Type:** Functional	**PUC:** Search client
Description: In case of no client available for a certain eatery and the customer is in a hurry, the app shall suggest other eateries that have available client(s).		
Rationale: Help customers solve the problem actively and try to let them stay with the app and open it whenever it's possible, since data like retention rate and DAU, user experience and word-of-mouth are all important indicators of success.		
Originator: Operation team		
Fit Criterion: At the bottom of the eatery's information page, if there is no available client, a pop-up box showing "other options" shall appear and redirect customers to other eateries that have available client(s).		
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL06 in terms of UX design (these 2 solutions are for the same problem, how to present them to the user without confusion?).	

Requirement ID: ZL08	**Requirement Type:** Functional	**PUC:** Search client
Description: In case of no eatery available, there shall be a suggestion to alternative dining options.		
Rationale: Help customers solve the problem and make them feel compassioned.		
Originator: Operation team		
Fit Criterion: If search result is empty, ● the app shall provide a list of alternative options like nearest vending machine, nearest grocery/convenience store, etc.; ● each option has opening hours, phone number, address and one sentence description; ● when a customer clicks one of the options, the app shall present the navigation on a map.		

Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL01 since the results are technically not eateries.

Requirement ID: ZL09	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall design function(s) for special notice or advertisement on search result list.		
Rationale: Leave opportunities for emergency, special situations (like COVID-19), promotion (like seasonal special) and monetization (advertisement) in order to better engage users.		
Originator: Operation team and eatery management		
Fit Criterion: On top of the search result list, there shall be a banner/textbox that only exists when there is alert/warning, promotion or other information.		
Supporting Materials: A2 Vision	**Conflicts:** May be disturbing and annoying for some users.	

Requirement ID: ZL10	**Requirement Type:** Functional	**PUC:** Search client
Description: The app shall notice the users (customer and client) of the best time to order.		
Rationale: Help customers avoid missing lunch/dinner time and make them feel compassioned.		
Originator: Customer, client and operation team		
Fit Criterion: The app shall: ● show notice/alert for eateries that are closing soon; ● push notification when eateries and clients are available, but the user has not ordered yet.		
Supporting Materials: A2 Vision	**Conflicts:** May be disturbing and annoying for some users.	

Requirement ID: *RA01*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine domain is Online Serviced Shared Economy where users highly trust and depend on ratings from other online community members. SkiptheLine app shall display a 5-star rating system next to Client(s) to help Customers pick and choose Client(s) faster. Client(s) with higher ratings will be programmed to be ranked at top.		
Rationale: In order help resolve Customer decision-making faster on choosing Client(s) and placing food order.		
Originator: Development Team		
Fit Criterion: Given 5-star rating option, Customers will choose Client(s) with best rating and decide to place food orders with them within 10 seconds.		
Supporting Materials: A1 & A2 Vision Business Use Cases: 2,3	**Conflicts: TBD**	

Requirement ID: *RA02*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall use External Location Interface to help Customers view Client(s) location real-time and compute how far Client(s) are from Customer. Client(s) nearest to the Customer (going to Customers desired Eatery) will be ranked at top. The app shall display distance and rating system next to the Client.		
Rationale: In order for Customers to choose and place food order with Client(s) nearest to them.		
Originator: Development Team		
Fit Criterion: Given the External location interface, Customers will choose Client closest to them and decide to place food orders with them within 10 seconds.		
Supporting Materials: A1 & A2 Vision Context Diagram: SkiptheLine uses External Location Interface Business Use Cases: 2,3,4	**Conflicts: TBD**	

Requirement ID: *RA03*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall use Applozic's Instant Messenger, where Customers and Clients can chat with one another regarding Food Orders, Payment methods, Confirmation and finding one another.		
Rationale: In order to fasten the process of placing food order, Customers will directly message Clients.		
Originator: Development Team		
Fit Criterion: Given Customer finds Client(s) going to their desired Eatery through the Instant Messenger, Customer cam message Client food order details and payment method and within 30 seconds can get food order confirmation from Client.		
Supporting Materials: A2 Vision: Strategic Dependency Model Business Use Cases: 3,4	**Conflicts: TBD**	

Requirement ID: *RA04*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have an Alert Message Box appear if the Client is full in capacity of taking food orders. Client(s) are only allowed to take up to 3 orders. Also, this will help save Customers time in waiting for Clients to respond. SkiptheLine will block any Customers contacting Clients with more than 3 orders.		
Rationale: To ensure Clients are not taking more than 3 Customer Food Orders.		
Originator: Development Team		
Fit Criterion: Given Customer searches for Clients going to their desired Eatery when Customer taps on a Client; an Alert Message Box appears stating Client is full in capacity of taking orders.		

Supporting Materials: A2 Vision Business Use Cases: 3	**Conflicts: TBD**

Requirement ID: *RA05*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have Block feature for SkiptheLine Instant Messenger. This functionality will help App Users remove and block Users they don't want to receive messages from. Also, if an App User has blocked another User than the blocked User cannot rate them.		
Rationale: App Users can filter-out/Block Users from their Messenger they had bad experience with		
Originator: Administration Team and/or Operation Team		
Fit Criterion: Given a Customer did not have a good experience in the past with a Client and when they see the same Client then the Customer can instantly (within 5 seconds) block the Client.		
Supporting Materials: A2 Vision Business Use Cases: 3,4	**Conflicts: TBD**	

Requirement ID: *RA06*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have Live Chat to resolve Customer and Client issues. The Live Chat will have the Administration team available 24/7 providing support.		
Rationale: To ensure App Users have a platform to share concerns/problems timely by Administration.		
Originator: Administration Team		
Fit Criterion: Given that the Customers while using Instant Messenger are experiencing technical issues; the Customer messages Administration team for help through Live Chat. Most likely the Administration team will then reply back within 5 mins.		
Supporting Materials: A2 Vision Business Use Cases: 1,2,3,4	**Conflicts: TBD**	

Requirement ID: *RA07*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have Notification functionality to update both Customer and Client for new message(s). This functionality will help both Customer and Client get informed of new messages without continuously checking the phone. There may be increased amounts of wait time between Customer and Client during the process of placing order and confirming order.		
Rationale: Help resolve any time gaps in communication between Customer and Client		

Originator: Development Team	
Fit Criterion: Given the Customer has high speed internet and messaged Client requesting Food Order, When the Client messages Customer accepting their request then the Customer receives instant new message notification within 5 seconds.	
Supporting Materials: A2 Vision Business Use Cases: 3,4	**Conflicts: TBD**

Requirement ID: *RA08*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have Photo Sharing functionality in SkiptheLine Instant Messenger. This functionality is super useful for Clients sharing a photo of food order received at Eatery. Or if there is any change in the menu, the Client can share menu with Customer by taking a picture.		
Rationale: Create an effective and engaging way of feedback/communication between Customer and Client.		
Originator: Development Team		
Fit Criterion: Given Customers requested order was unavailable, the Client sends the Customer a picture of the Eateries menu. The Customer then messages back the Client another food order request and this process takes around 2 mins.		
Supporting Materials: A2 Vision Business Use Cases: 3,4	**Conflicts: TBD**	

Requirement ID: *RA09*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have Report feature for SkiptheLine Instant Messenger. Customers and Clients can report any sort of misuse language on Instant Messenger between both App Users to Administration Team.		
Rationale: For both Customers and Clients who experience any sort of misuse, can immediately take action and report the misuse event (i.e. Abusive Language...)		
Originator: Administration Team		
Fit Criterion: Given Customer requests food order, however the Client happens to send Customer abusive messages. The Client reports the event by clicking on the Report feature, and leaves a brief message and Blocks the Client; this process takes within 2 minutes.		
Supporting Materials: A2 Vision Business Use Cases: 3,4,5	**Conflicts: TBD**	

Requirement ID: *RA10*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall have a 'Subscribe' button, where Customers can Subscribe to Eateries or Client(s) near them. When a Client updates their status, going to an Eatery; Both Eateries and/or Clients subscribed Customers nearby, will be notified instantly.		
Rationale: Customers may have favorite Eatery Restaurants or Client(s) that they want to keep up with and follow.		
Originator: Development Team		
Fit Criterion: Given a Customer really likes Tim Hortons coffee, whenever a Client updates their status (of going to that eatery) then nearby Customers receive instant notification about that Client. The Customer then plans on messaging the Client for an order (under 2 mins).		
Supporting Materials: A2 Vision Business Use Cases: 3	**Conflicts: TBD**	

Requirement ID: *RA11*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall save all Customer and Client Food Order Transactions into cloud database. This will help resolve any tractability and accountability issues (between Customer or Client).		
Rationale: For record keeping of Food Order Transactions between Customers and Clients.		
Originator: Operations Team		
Fit Criterion: Given a Customer liked the service of a Client and wants to subscribe and get notifications from them; however, the Customer forgot the Client's name. The Customer then searches past transactions and finds the Client's name and Subscribes with them. This process took the Client around 3 mins.		
Supporting Materials: A2 Vision Business Use Cases: 3	**Conflicts: TBD**	

Requirement ID: DZ01	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall send a notification to the Customer once the Client has purchased the order from the eatery.		
Rationale: The Customer should know when to expect the Client to be on their way and be prepared to help the Client to locate and identify themselves.		
Originator: Client and Customer		
Fit Criterion: On the 'order information' page, the Client shall be able to click on a button to acknowledge that he/she has purchased the order. Once the Client clicked on the button, the App shall send the Customer an automatic notification that the Client has successfully purchased their food.		

Supporting Materials: A2 Vision	**Conflicts:**

Requirement ID: DZ02	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow the Client to access the Customer's profile photo and vice versa.		
Rationale: The Client and the Customer should be able to use each other's photo to help identify each other.		
Originator: Client and Customer		
Fit Criterion: Once the Client purchased the food and the App notified the Customer, the Client and the Customer shall be able to click on each other's profile on the 'order information' page and zoom in on each other's profile photo (if they have one).		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ03	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall enable a GPS tracking network to update the location of the Client and Customer.		
Rationale: The Client and the Customer should be able to locate each other on the virtual map on the app.		
Originator: Client and Customer		
Fit Criterion: Once the Client has purchased the order, the Client and the Customer should be able to see each other's live location on the virtual map on the App. The location shall be updated instantly.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ04	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow Users to adjust their location on the virtual map manually.		
Rationale: In case the GPS tracking feature fails to locate the Customer or the Client accurately or timely, the Customer or the Client should be able to adjust their location to help each other find themselves more easily.		
Originator: Customer and Client		
Fit Criterion: On the virtual map, the Client and the Customer should be able to move the symbol representing their current location. The new location shall be updated on each other's virtual map page.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ05	**Requirement Type:** Functional	**PUC:** Search for Customer

Description: The App shall allow the users to start a text conversation related to an order.	
Rationale: The Customer and Client should be able to communicate using the app to help each other locate themselves.	
Originator: Customer and Client	
Fit Criterion: On the "order information' page, there should be a textbox for the Customer and the Client to type in their conversations under the 'chat' feature. The other person will receive a notification of the message and the order ID related to it.	
Supporting Materials: A2 Vision	**Conflicts:**

Requirement ID: DZ06	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow the users to send pictures during a conversation related to an order.		
Rationale: The Customer and Client should be able to send each other photos of their surroundings to help each other find themselves.		
Originator: Customer and Client		
Fit Criterion: In the textbox under that 'chat' feature, there should be an option to upload pictures from the gallery and an option to take a photo.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ07	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow users to start a voice chat during a conversation related to an order.		
Rationale: The Customer and Client may want to call each other to help each other find themselves.		
Originator: Customer and Client		
Fit Criterion: Under the 'chat' feature, there should be an option to start a voice chat. The other person will receive a notification whether to accept the voice chat.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ08	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow users to save the conversations related to specific orders.		
Rationale: The Customer and Client may want to save the conversations as evidence of the other person's non-responses and/or aggressive languages.		
Originator: Customer and Client		
Fit Criterion:		

Under the 'chat' feature, there should be an option to send the transcript to the user's designated email address, along with the order number and both Client and Customer's IDs.	
Supporting Materials: A2 Vision	**Conflicts:**

Requirement ID: DZ09	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall allow users to report aggressive languages, verbal harassment and discriminations during conversations to the system administrator.		
Rationale: The App shall ensure the users rights are protected when interacting with other users, hence disrespectful behaviours should be reported for investigations.		
Originator: Client, Customer, Operation Team		
Fit Criterion: There should be a 'report an issue' option under the 'help' section of the App, which allows the user to enter the order number, attach the chat transcript, and describe the issues they encountered.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ10	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The system shall allow system administrators to suspend certain users' accounts if they can confirm that the users have violated the user agreement by verbally abusing other users during conversations.		
Rationale: The system shall protect all users' rights when using the App, hence no harassment or discriminative behaviours should be tolerated.		
Originator: Operation Team		
Fit Criterion: After a user submitted a 'report an issue' form, a system administrator should be able to access the order number, both users' IDs and the conversation history. The system administrator should also be able to contact the users using their IDs to investigate further. The interface should have an option for the system administrator to send warnings and suspend a user's account if the user is deemed to have violated the user agreement.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: FK01	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application shall contain four methods of payment: Cash, e-Transfer, Inbu, T-bucks deposit using credit card. This must be present on the user profile section of the app. The application must require app users to select their desired methods of receiving payment as part of their user profile, if they are willing to take on the role of client. App users who leave this field blank will not be recognized as potential clients by the app.		
Rationale: Some clients may not accept certain methods of payments and this prevents conflict		

between app users who can only pay using a certain method and those who can only accept a certain method
Originator: App Operations Team
Fit Criterion: Given that the client will use the app to deliver food customers, they will select at least 1 method of payment as their desired form of payment in their user profile.

Requirement ID: FK02	**Requirement Type:** Functional	**PUC:** Making Payment
Description: When a customer selects an app user who is headed to their eatery of choice, the application shall cause an "accept/deny order" notification pops up for App users.		
Rationale: This allows for the app user to be separated into customer and client roles for record keeping and makes it clear who will be paying whom		
Originator: App Developer		
Fit Criterion: given that a potential client is selected by a customer, the client shall see the notification within 10 seconds of being selected.		
Supporting Materials: Business use case 2,3 Definitions and assumptions Strategic Dependency I*	**Conflicts:**	

Requirement ID: FK03	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application shall contain an 'order in progress page' visible to both the client and the customer once the food order is initiated and the customer has successfully located a client heading to the customer's desired eatery. The 'order in progress' interface shall be slightly different for clients and customers		
Rationale: This page will allow customers to chat with clients, track the location of the client on the google maps plugin, make payments, and accept payments.		
Originator: Customer		
Fit Criterion: Given that the customer selects their desired client and the client accepts the customer's order, the 'order in progress' page must be visible to both parties within 3 seconds of the acceptance of the order by the client		
Supporting Materials: Business use case 2,3 Google maps API terms and conditions Terms and definitions	**Conflicts:**	

Requirement ID: FK04	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The customer 'order in progress' page shall contain a 'Make Payment' button which takes the customer to a new page on the app containing fields such as "order amount", "tips and gratuities", payment options and a "submit payment" button. The tips and gratuities section will have tip		

denominations from 5%, 10% , 15% of the bill or customers will have the option to manually type in their desired amount. If the payment options of in-app payment system is selected, the customer will be taken to the payment page to continue the transactions. If cash or eTransfer options are selected, a 'submit payment' option occurs and customers can press it if the payment has been made.
Rationale: Since the food is not being ordered via the app, customers will have to manually insert the amount from their order into the 'order amount' field and manually select any tips or gratuities options. Since Cash and eTransfer options occur outside of the app, customers just click 'submit payment' to end transaction
Originator: Application Operations Team
Fit Criterion: Given that the customer has an order in progress and the client arrives with the customer food, customer clicks on the 'Make Payment' page and fills in all the appropriate. information

Supporting Materials:	**Conflicts:**
Definitions and assumptions Context Diagram Business use case 1,2,3 Square checkout integration guide Transact Campus Documentation	

Requirement ID: FK05	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application shall contain an 'order complete' button for the client to press once he has received his payment. The records will show 'payment pending' until the client clicks on the 'order complete' button. An order number shall be produced and the order number shall link to a cloud database to allow client and customer to view chat history. Nothing except the order number is saved on the app user's phone.		
Rationale: to be able to log the end of a transaction in the cloud database, end the transaction in the app and store transaction information in the cloud database		
Originator: App Developer		
Fit Criterion: When a client initially accepts an order from a customer, the complete order button pops up at the top of the app screen within 5 seconds of order confirmation. Transaction details should be uploaded into the cloud within 5 seconds of the press of the button.		

Supporting Materials: Business cases 2,3 Definitions and assumptions Context Diagram	**Conflicts:**

Requirement ID: FK06	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application shall contain an order history page where the chat history, the payment method used, and an optional picture of the physical receipt can be stored under a unique order number. Nothing except the order number is saved on the app user's phone.		
Rationale: Customers and clients can both see their transaction histories and this transparency can mitigate potential conflict and allow customers to keep track of their orders		

Originator: App User	
Fit Criterion: Given that the user is on the app and clicks on the app menu, The app menu contains order history page	
Supporting Materials: Business use case 2,3,4 Definitions and assumptions	**Conflicts:**

Requirement ID: FK07	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application must contain a feature to upload a picture of the physical receipt into the cloud database with the transaction number attached as a tag.		
Rationale: for recordkeeping purposes, if client/customer conflict arises, receipt can serve as evidence		
Originator: App operations team		
Fit Criterion: Given that the app user accepts a customer's order and takes up the role of client, an 'upload receipt' button appears as an option in the chat bar menu		
Supporting Materials: Business use case 3,4 Definitions and assumptions	**Conflicts:**	

Requirement ID: FK08	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application must be able to integrate with Square API to accept credit card payments. The application must be able to take customer to t-bucks 'guest' deposit page if client prefers being paid in T-bucks		
Rationale: to add flexibility to payment options by allowing for credit card payments while keeping information secure by conducting the payment through the in-app payment system l. Uoft students pay prefer being paid in T-bucks for a variety of reasons		
Originator: Client		
Fit Criterion: Given that the customer selects the option to pay using the in-app payment system, the customer is taken to a payment screen to input their name and credit card information within 2 seconds of selecting the in-app payment option		
Supporting Materials: Business Use case 1,2,3 Vision Onion Model Definitions and Assumptions Square API guide Transact-campus Documentation	**Conflicts:**	

Requirement ID: FK09	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The app must not provide customers with the option to save their credit card information		

Rationale: This slight inconvenience will discourage customers from using the built-in payment system and encourage the use of peer to peer payment methods.	
Originator: Customer	
Fit Criterion: Given that the customer selects the option to pay with the built-in system, the customer is taken to the payment page on the app where they are asked to input their name, credit card number and expiry date, with no option to save the information for future use.	
Supporting Materials: Definitions and Assumption Square developers guide	**Conflicts:** FK11

Requirement ID: FK10	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application must display a liability waiver message to let customers and clients know that investigations cannot occur for payments made in cash or through Interac-e-Transfer		
Rationale: Since these actions happen outside the app, there is not a way to verify the validity of the information if a complaint is lodged		
Originator: App Operations Team		
Fit Criterion: Given that a customer selects pay-in-cash or e-transfer option, a notification pops up telling them that a complaint lodge about this order with regards to funds will not be investigated. When the client selects cash or e-transfer as their desired method of payment on their user profile, they will receive the same message.		
Supporting Materials: Context Diagram Definitions and assumptions	**Conflicts:** FK11	

10 NON-FUNCTIONAL REQUIREMENTS

Requirement ID: *CAA11*	**Requirement Type:** Performance	**PUC:** Create a SkiptheLine profile
Description: The system shall be robust and not crash due to overload or simultaneous login or registration		
Rationale: To ensure that users are able to login or create a profile at any time and during peak periods of the application.		
Originator: Developers		
Fit Criterion: There must be 95% efficient login or registration of users during peak periods of the application. The application shall generate a report back to system administrators within 2 seconds in the event of a crash.		

Supporting Materials: BUC #1, PUC Table, SD Model (A2 Vision).	**Conflicts:** Might conflict with 19 due to capacity required

Requirement ID: *CAA12*	**Requirement Type:** Usability	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure that the profile creation process is straightforward and intuitive for the user.		
Rationale: To ensure that new users are able to quickly and easily create a profile within 1minute of the entire profile creation process and begin using the SkiptheLine application.		
Originator: Developers		
Fit Criterion: The profile creation process shall be straightforward and intuitive for users of all ages, including elderly individuals and those with minimal experience in mobile applications. User must be able to create their profile within 1 minute, and 95% of survey respondents shall report that the profile creation process was easy to complete.		
Supporting Materials: BUC#1, PUC Table, SD Model (A2 Vision).	**Conflicts:** Might conflict 16	

Requirement ID: *CAA13*	**Requirement Type:** Understandability	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure that information contained in both the Terms of Use and Privacy Policy statements is presented in a clear and understandable way.		
Rationale: To ensure that users feel informed and comfortable in their use of SkiptheLine and how their data will be used.		
Originator: Legal team		
Fit Criterion: At least 70% of survey respondents shall report feeling well informed about these policies and 90% shall feel that the SkiptheLine application is trustworthy.		
Supporting Materials: BUC#1, PUC Table, Onion Diagram, SD Model (A2 Vision).	**Conflicts:TBD**	

Requirement ID: *CAA14*	**Requirement Type:** Accessibility	**PUC:** Create a SkiptheLine profile
Description: The system shall be accessible to people with vision impairments.		
Rationale: To ensure accessibility and ease of use to all types of users, including those with impaired vision.		
Originator: Developers		
Fit Criterion: The product shall include industry standard accessibility features, including those that allow the user to increase font size, magnify portions of the screen, and implement a ‘dark mode’.		

Supporting Materials: BUC #1, PUC Table, Onion Diagram, SD Model(A2 Vision).	**Conflicts:TBD**

Requirement ID: *CAA15*	**Requirement Type:** Privacy	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure that existing users are actively notified of changes to the SkiptheLine application's Terms of Use and Privacy Policy.		
Rationale: To ensure that users feel informed and comfortable in their use of SkiptheLine and how their data will be used.		
Originator: Legal team		
Fit Criterion: The legal team shall create a statement for distribution 21 days prior to any changes being made. Users shall be presented with a notification on the application homepage regarding these changes and be sent an email 14 days prior to changes coming into effect.		
Supporting Materials: BUC #1,2,3,4,5 PUC Table, Onion Diagram (A2 Vision).	**Conflicts:TBD**	

Requirement ID: *CAA16*	**Requirement Type:** Privacy	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure that user personal information including passwords are stored securely within the SkiptheLine database using advanced techniques like password hashing.		
Rationale: To ensure that that user's personal information is secure and that damage to SkiptheLine is minimized in the event that the system is compromised.		
Originator: Developers		
Fit Criterion: Personal information including passwords shall be safely stored using hash methods, masking or similar. No passwords shall be stored in plain text.		
Supporting Materials: BUC #1, PUC Table (A2 Vision), Personal Information Protection and Electronic Documents Act (PIPEDA).	**Conflicts: TBD**	

Requirement ID: *CAA17*	**Requirement Type:** Localization	**PUC:** Create a SkiptheLine profile
Description: The language of the system shall be in the recognised official languages spoken in Canada and must have a grammar and spelling checker.		
Rationale: To ensure that the users are able to communicate in the system using the common official languages understandable to everyone within Canada.		
Originator: Developers		

Fit Criterion: All instructions for user data entry and data saved in the system must be readable in both English and French with grammar and spelling checker that facilitates correction of sentences and words in their right form.	
Supporting Materials: BUC #1, PUC Table, Onion Diagram (A2 Vision).	**Conflicts:TBD**

Requirement ID: *CAA18*	**Requirement Type:** Integrity	**PUC:** Create a SkiptheLine profile
Description: The SkiptheLine system shall delete user profiles that have not been logged into for 365 consecutive days and provide 14 days' notice to the user associated with that profile via email with the option to prevent deletion.		
Rationale: To ensure efficiency of data storage and integrity of data within the SkiptheLine database.		
Originator: System Administrators		
Fit Criterion: All data associated with profiles not logged into for over 365 days shall be deleted from the system database completely. The system shall notify all users whose profiles are subject to deletion receive 14 days' notice via email 95% of the time. No user profile shall be deleted should that user does not receive 14 days' notice.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).		**Conflicts:TBD**

Requirement ID: *CAA19*	**Requirement Type:** Capacity	**PUC:** Create a SkiptheLine profile
Description: The system shall be capable of maintaining up to 200,000 active profiles effectively and efficiently.		
Rationale: To ensure that the SkiptheLine application is available to a wide customer base, while also ensuring the efficiency of data storage and integrity of data within the SkiptheLine database.		
Originator: System Administrators		
Fit Criterion: 200,000 active profiles shall be maintained simultaneously with no impact on application performance and quality. User reviews shall not indicate lagging application performance as the system nears its capacity.		
Supporting Materials: BUC #1, PUC Table (A2 Vision).		**Conflicts:TBD**

Requirement ID: *CAA20*	**Requirement Type:** Security	**PUC:** Create a SkiptheLine profile
Description: The system shall record each unsuccessful attempt to login and require the user to reset their password after 7 unsuccessful attempts.		
Rationale: To ensure the privacy of user personal information and other sensitive information like payment options.		

Originator: System Administrators	
Fit Criterion: The user shall receive a reset notice following 7 unsuccessful attempts from the system administrator, outlining the steps required to reset their password.	
Supporting Materials: BUC #1, PUC Table, SD Model(A2 Vision).	**Conflicts:TBD**

Requirement ID: ZL11	**Requirement Type:** performance	**PUC:** Search client
Description: The list of available eateries and clients shall be updated on a real-time basis.		
Rationale: To avoid delays or mismatch between customer, client and eatery.		
Originator: Customer, client and operation team		
Fit Criterion: The app shall refresh in a very short time interval or even as frequently as possible, depending on the technical review from the development team.		
Supporting Materials: A2 Vision	**Conflicts:** May conflict against requirements about low battery usage, device compatibility, budget and technology constraints, etc.	

Requirement ID: ZL12	**Requirement Type:** operational	**PUC:** Search client
Description: The app shall highlight recommendations, such as popular eateries and popular menu items.		
Rationale: To facilitate users' decision-making process and enhance their user experience.		
Originator: Customer, client and operation team		
Fit Criterion: The app shall: • generate recommendations based on existing user data; • retrieve and analyze data from external service providers such as Google and Yelp.		
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL01 and ZL02 since the "recommendation" might be biased, and against ZL14 since the result might be not welcomed by some group, and ZL17 for the recommendation here is not personalized.	

Requirement ID: ZL13	**Requirement Type:** maintainability	**PUC:** Search client
Description: The app shall allow users (customer and client) to report the incorrect or incomplete information about any eateries and menu items.		
Rationale: To maintain the service in a collaborative way, involve all the users and complement the operation team's work.		
Originator: Operation team		
Fit Criterion: The app shall have a small button in a not-so-obvious location on both the search result page and eatery information page for users to fill a form and send the correction suggestion.		

Supporting Materials: A2 Vision	**Conflicts:** TBD

Requirement ID: ZL14	**Requirement Type:** cultural	**PUC:** Search client
Description: The app shall allow users (customer and client) to select eateries according to their dining preferences in terms of culture, religion, physical condition, etc.		
Rationale: To show caring and respect to the diversified community.		
Originator: Operation team		
Fit Criterion: The search result list of eateries shall be able to be filtered through preferences like “vegetarian”, “halal”, “gluten free” etc.		
Supporting Materials: U of T Food Service – Food Standards[1]	**Conflicts:** May conflict against ZL18 since adding filtered may complicate the usage.	

Requirement ID: ZL15	**Requirement Type:** appearance	**PUC:** Search client
Description: The search result list shall be easy to read, understand and attractive to U of T members, primarily students.		
Rationale: To ensure that the appearance conforms to the usual UX design principles and the preference of young users.		
Originator: Customer, client, development team and operation team		
Fit Criterion: The UX design of the app shall consider: ● Visibility of system status (use eye-catching but elegant way to indicate important status of the client searching process); ● Aesthetic and minimalist design (use colours and other design elements that are suitable for dining time).		
Supporting Materials: “10 Usability Heuristics for User Interface Design” by Jakob Nielsen[2]	**Conflicts:** “Aesthetic and minimalist design” may conflict against requirements for “User control and freedom” in ZL16.	

Requirement ID: ZL16	**Requirement Type:** ease of use	**PUC:** Search client
Description: The app shall make the functions and operations easy to understand so that users can learn intuitively.		
Rationale: Need to avoid delays or mismatch between customer, client and eatery.		
Originator: Customer, client, development team and operation team		
Fit Criterion: The UX design of the app shall consider: ● User control and freedom; ● Consistency and standard; ● Error prevention;		

[1] https://ueat.utoronto.ca/everythingfood/food-standards/

[2] https://www.nngroup.com/articles/ten-usability-heuristics/

• Recognition rather than recall.	
Supporting Materials: "10 Usability Heuristics for User Interface Design" by Jakob Nielsen	**Conflicts:** May conflict against ZL18 since user control and too much freedom may cause the app less "foolproof".

Requirement ID: ZL17	**Requirement Type:** personalization	**PUC:** Search client
Description: The app shall provide features with personalized touch.		
Rationale: To provide considerate service on a higher and smarter level.		
Originator: Operation team		
Fit Criterion: The app shall: • remember individual user's sorting and filtering preferences (and allow users to turn off auto-remembering feature in the general setting); • make use of individual user's historical data and provide options like "reorder shortcut" from a previous order; • make use of "big data", provide suggestions like "people like you also like these eateries".		
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL20 since more data (in terms of volume and types) can produce better results, but may cause infringement to data privacy.	

Requirement ID: ZL18	**Requirement Type:** convenience	**PUC:** Search client
Description: The app shall reduce the steps for a user (customer or client) to find the desired result.		
Rationale: To prevent users from quitting/churning due to complicated conversion funnel.		
Originator: Operation team		
Fit Criterion: On top of what's included in ZL11, ZL12 and ZL17, the app shall consider		
Supporting Materials: AARRR framework by Dave McClure[3].	**Conflicts:** May conflict against ZL14 and ZL16 in early stages.	

Requirement ID: ZL19	**Requirement Type:** reliability & precision	**PUC:** Search client
Description: The app shall always locate the users (customer and client) accurately, especially in the urban setting of campus and inside the buildings (which may cause technical difficulties).		
Rationale: Need to minimize errors and mistakes when customers and clients look for each other and enhance user experience.		
Originator: Customer, client and development team.		

[3] AARRR Framework- Metrics That Let Your StartUp Sound Like A Pirate Ship: https://medium.com/@ms.mbalke/aarrr-framework-metrics-that-let-your-startup-sound-like-a-pirate-ship-e91d4082994b

Fit Criterion: The app shall: ● retrieve users' location on real time basis; ● use cutting edge technologies that realize precise locating; ● suggest users to help facilitate locating by using Wi-Fi, etc.; ● include functions that allow users to describe where they are in text format.	
Supporting Materials: A2 Vision	**Conflicts:** May conflict against ZL20, and requirements about low battery usage, device compatibility, budget and technology constraints, etc.

Requirement ID: ZL20	**Requirement Type:** privacy	**PUC:** Search client
Description: The app shall comply with the law and protect individual privacy of users.		
Rationale: Need to do good things for the community, establish positive public image, avoid user dissatisfaction and complaint, and even lawsuit.		
Originator: Operation team		
Fit Criterion: The app shall follow the 10 fair information principles in Schedule 1 of PIPEDA, especially: accountability, identifying purposes, consent, limiting collection, limiting user, disclosure and retention, etc.		
Supporting Materials: Personal Information Protection and Electronic Documents Act (PIPEDA)[4].	**Conflicts:** May conflict against ZL17, ZL18 and ZL19.	

Requirement ID: *RA12*	**Requirement Type:** Personalization and Internationalization	**PUC:** Place Food Order
Description: SkiptheLine app shall allow Customers and Clients to customize and change app settings according to their desires/requirements · Personal Configuration options (i.e. change language of app) · Customer can Subscribe to their favorite Client(s) and/or Eateries · App Users have Live Chat available to discuss concerns. · App Users can block and report Users · App Users can rate each other · App shall use symbols that are internationally recognized/understood		
Rationale: To ensure App Users interests/requirements are met.		
Originator: App Users (Customer and Clients)		
Fit Criterion: Given App Users now dislikes Tim Hortons coffee and wants to subscribe with Starbucks. After the App User then changes the language of the app into in French. The App User was able to customize all app settings while waiting for their shuttle bus in less than 2 minutes.		
Supporting Materials:	**Conflicts: TBD**	

[4] PIPEDA in brief: https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/pipeda_brief/#_h2

A1 & A2 Vision Business Use Cases: 1,2,3,4	

Requirement ID: *RA13*	**Requirement Type:** Ease of Use	**PUC:** Place Food Order
Description: SkiptheLine App shall be easy for Customers and Clients to use. The app will need no tutorials or manual for instructions on how to use app. Since SkiptheLine is service based mobile app: Customers usually use it when they are hungry and Clients use it when they are willing to provide service. This app is not used as pass time, and it serves a purpose. App Users are often in a rush and would want the app to be easily understood.		
Rationale: Customers are hungry and want food and Clients are constantly traveling and using the app; Hence, the app features should be simple and easy to navigate.		
Originator: Development Team		
Fit Criterion: Given there are many Clients going to Pizza Pizza. At Customers first attempt with SkiptheLine mobile app, the Customer was able to navigate and find Clients going to their desired eatery within 1 min.		
Supporting Materials: A2 Vision: Strategic Dependency Model Business Use Cases: 1,2,3,4,5	**Conflicts: TBD**	

Requirement ID: *RA14*	**Requirement Type:** Availability	**PUC:** Place Food Order
Description: SkiptheLine app is available 23 hours a day, 7 days a week and 365 days per year; as SkiptheLine purpose is to enhance food consumption experience better. However, most Eatery Restaurants on campus are open on certain hours, the app will display alternatives on campus (i.e. Vending Machines nearby). The app may have scheduled downtime or for technical reasons which is why the expected available hours for the app is 23 hours a day.		
Rationale: Since students have different food consumption timing (i.e. exam time, hectic schedules, overnight shifts) and are always looking for food options; SkiptheLine app tries to be available 24/7.		
Originator: Operation Team		
Fit Criterion: During exam and midterms, SkiptheLine app will be available 24 hours a day, 7 days a week; running 99% of the time without major technical issues.		
Supporting Materials: A1 & A2 Vision Business Use Cases: 1,2,3,4,5	**Conflicts: TBD**	

Requirement ID: *RA15*	**Requirement Type:** Scalability	**PUC:** Place Food Order
Description: SkiptheLine shall be able to maintain Food Order Transactions for 2020-2021 Fall/Winter/Summer semesters.		

Rationale: To ensure SkiptheLine has the capability to store all data information between Customer and Client and handle app growth	
Originator: Sponsors and Owner	
Fit Criterion: Given App Users data is expected to double for 2021-2022 Fall/Winter/Summer semesters. When the year 2021 comes, then SkiptheLine app should be able to migrate/extend into a larger cloud-based storage and is capable of storing double the amount of data.	
Supporting Materials: A1 & A2 Vision Business Use Cases: 3	**Conflicts: TBD**

Requirement ID: *RA16*	**Requirement Type:** Appearance	**PUC:** Place Food Order
Description: SkiptheLine app shall attract to University level students. Instant Messenger will have the spirit of UofT students and have standard chat features.		
Rationale: To ensure SkiptheLine Instant Messenger appearance and spirit is suitable and supports the most effective way to communicate between Customer and Client for University level students.		
Originator: Owner		
Fit Criterion: Customers will be attracted to the app and easily navigate and request food order within 3 mins of use. Also, Clients will be able to adapt quickly and fulfill Customer food orders.		
Supporting Materials: A1 & A2 Vision Business Use Cases: 3,4,5	**Conflicts: TBD**	

Requirement ID: *RA17*	**Requirement Type:** Convenience	**PUC:** Place Food Order
Description: Customers can find Clients going to their desired Eatery and place food orders with them. Customers give Clients tips in exchange for their service. Both Customer and Client can benefit from the SkiptheLine app.		
Rationale: The app shall help Customers place food order through mobile and without physically going to the Eatery Restaurant		
Originator: App Users		
Fit Criterion: Given Clients can receive tips in exchange of helping another student, then the Clients will be more compelled to help in the future.		
Supporting Materials: A1 & A2 Vision Business Use Cases: 1,2,3,4,5	**Conflicts: TBD**	

Requirement ID: *RA18*	**Requirement Type:** Speed	**PUC:** Place Food Order
Description: SkiptheLine app shall use mainly Instant Messenger and Messenger notifications for the PUC: place food order. While navigating, the response time of SkiptheLine app will be maximum 3 seconds (includes retrieving Customer to Client distance)		
Rationale: To ensure there are no interruptions and the App Users could navigate the app without any long pauses (specially when communicating, and waiting for response)		
Originator: Development Team		
Fit Criterion: Given App Users have good internet connection, the accepted response time for the app is 3 seconds (on every change that occurs).		
Supporting Materials: A2 Vision Business Use Cases: 3	**Conflicts: TBD**	

Requirement ID: *RA19*	**Requirement Type:** Precision and Accuracy	**PUC:** Place Food Order
Description: SkiptheLine app will store all Client rating and location (using External Location Interface: which provides real-time location difference between Customer and Client) information into the database. The App will then retrieve information from the database and will display it for the Customers. SkiptheLine shall accurately round computed distance (between Customer and Client) and ratings. While choosing the best Client, Customer needs to have accuracy in results produced on Available Clients page.		
Rationale: In order for Customers to choose best Client (best rating and Client nearest to them), Customer needs to have accurate results produced.		
Originator: Development Team		
Fit Criterion: Given Customer search result list, rating results need to be accurate. If ratings have decimal points then the ratings will be rounded up (not necessary for location).		
Supporting Materials: A1 & A2 Vision Business Use Cases: 2,3	**Conflicts: TBD**	

Requirement ID: *RA20*	**Requirement Type:** Capacity	**PUC:** Place Food Order
Description: For 2019-2020 University of Toronto had in total around 93 thousand students. The app shall deal with Food Order Transactions data (between Customer and Client), Available Clients data (along with location and rating data) and Messenger data.		
Rationale: To ensure SkiptheLine app is capable of handling data during rush-hours (Lunch Time specifically)		
Originator: Development Team		

Fit Criterion: For the first month of release, SkiptheLine will be able to cater to 20 thousand App Users from 11 a.m. to 1 p.m.	
Supporting Materials: A1 & A2 Vision *Quick facts*. (University of Toronto, 2020) Business Use Cases: 2,3,4,5	**Conflicts: TBD**

Requirement ID: *RA21*	**Requirement Type:** Expected Physical Environment	**PUC:** Place Food Order
Description: SkiptheLine app shall be used by University of Toronto (all 3 campus) students primarily. The app will be used indoors (lectures, libraries, on-campus housing...) and/or outdoors (rain, sunny, hot, cold, server and all kinds of weather conditions).		
Rationale: To ensure all App Users can use SkiptheLine App under all intended environments listed in description.		
Originator: Development Team		
Fit Criterion: Given that Client receives food order from Customer and it happens to be very bright and sunny outdoors, SkiptheLine app UX/UI and appearance (colors, fonts) needs to be readable and professional (Specially for Clients outdoors and constantly walking).		
Supporting Materials: A1 & A2 Vision Context Diagram: SkiptheLine uses External Location Interface Business Use Cases: 1,2,3,4,5	**Conflicts: TBD**	

Requirement ID: DZ11	**Requirement Type:** Performance (Speed and Latency)	**PUC:** Search for Customer
Description: The App shall update the Customer and Client's locations constantly.		
Rationale: To inform Customer and Client each other's location at the moment accurately.		
Originator: Customer and Client		
Fit Criterion: The virtual map interface shall refresh users' location every second.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ12	**Requirement Type:** Performance (Precision or Accuracy)	**PUC:** Search for Customer
Description: The accuracy of users' locations shall be within 10 meters, or as accurate as the user's network capacity allows.		

Rationale: The App shall assist users to find their Customer/Client by providing their exact locations.	
Originator: Customer and Client	
Fit Criterion: The App shall use specialised GPS systems and location services to reduce errors.	
Supporting Materials: A2 Vision	**Conflicts:**

Requirement ID: DZ13	**Requirement Type:** Security (Privacy)	**PUC:** Search for Customer
Description: Only Client, Customer and Operation Team can see the order details, user IDs and photos and chat transcript.		
Rationale: The App shall protect user's data not to be shared with people/organizations unrelated to the order and the App.		
Originator: Client, Customer and Operation Team		
Fit Criterion: Not until the Client confirms that he/she has purchased the order, the Customer can see the Client's photo and user ID and start a conversation under the 'chat' feature. The Operation Team shall have access to the order details if the Client/Customer reports an issue related to the order.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ14	**Requirement Type:** Security (Privacy)	**PUC:** Search for Customer
Description: The App shall not share users' personal information data (location, phone number, email address, payment method) with third parties.		
Rationale: The App shall protect user's personal data not to be used for purposes other than placing orders on campus.		
Originator: Customer, Client		
Fit Criterion: User's personal information data should be encrypted and not to be used for advertising purposes.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ15	**Requirement Type:** Compliance (Legal)	**PUC:** Search for Customer
Description: Personal information shall be implemented so as to comply with the Data Protection Act.		
Rationale: To comply with the law so as to avoid later legal issues.		
Originator: Administration		
Fit Criterion: Consult lawyers' opinion that the product does not break any laws.		

Supporting Materials: The *Personal Information Protection and Electronic Documents Act* (Office of the Privacy Commissioner of Canada, 2020)	**Conflicts:**

Requirement ID: DZ16	**Requirement Type:** Maintainability (Adaptability)	**PUC:** Search for Customer
Description: The App is expected to run under iOS and Android		
Rationale: The App shall be accessible to nearly all students and people on UofT campus.		
Originator: Development Team		
Fit Criterion: The App shall have a version designed to run under iOS, and another version designed to run under Android.		
Supporting Materials: A2 Vision	**Conflicts:** The development of both versions of the app may result in a higher budget.	

Requirement ID: DZ17	**Requirement Type:** Operational	**PUC:** Search for Customer
Description: The App must interface with GPS tracking network and map service API.		
Rationale: The App shall integrate third-party GPS systems to keep the virtual map up-to-date.		
Originator: Development Team		
Fit Criterion: The App shall use GPS systems to update the buildings, streets, surroundings on the map in time, to help users to identify relative locations and reduce confusions.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: DZ18	**Requirement Type:** Performance (Availability)	**PUC:** Search for Customer
Description: The product shall be for use 24 hours per day, 365 days per year.		
Rationale: The App shall help users order food whenever they need to. Even outside eateries' opening hours, users should be able to check past orders and receipts, or report issues related to past orders.		
Originator: Development Team, Operation Team		
Fit Criterion: The App shall be available for use and does not fail at any time.		
Supporting Materials: A2 Vision	**Conflicts:** To operate and maintain the App 24 hours per day, 365 days per year could make the budget too high.	

Requirement ID: DZ19	**Requirement Type:** Usability	**PUC:** Search for Customer

Description: The App shall allow users to choose the form and frequency of communication.	
Rationale: The App shall cater to users' personal preferences in regards to notifications.	
Originator: Development Team	
Fit Criterion: Except critical communications (i.e. chat with the other user), users should be able to choose whether and how (email, messages, push notifications) they would like to receive notifications regarding order status.	
Supporting Materials: A2 Vision	**Conflicts:**

Requirement ID: DZ20	**Requirement Type:** Appearance	**PUC:** Search for Customer
Description: The interface design of the App shall be neat, distinctive and cater to most people on campus.		
Rationale: The App shall be easy to use for people with partial incapability (i.e. colour-blind)		
Originator: Development Team		
Fit Criterion: The App shall have high contrast between texts and background, and between symbols and background on the meap.		
Supporting Materials: A2 Vision	**Conflicts:**	

Requirement ID: FK11	**Requirement Type:** Ease of Use	**PUC:** Making Payment
Description: Payment methods are straightforward and easy to navigate through		
Rationale: The application makes it possible to easily transfer funds from customer to client		
Originator: Client		
Fit Criterion: Financial transactions are completed within 1 minute from the time the customer selects the payment option from the app menu till the time the client receives the payment, excluding cash and e-Transfer options		
Supporting Materials: Definitions and assumptions Strategic Dependency I*	**Conflicts:**	

Requirement ID: FK12	**Requirement Type:** Security	**PUC:** Making Payment
Description: No exchange of personal information occurs between client and customer, except for name. All personal account information is accessible by app user and with app user's permission, operations team personnel can access personal account information.		
Rationale: To prevent security breaches and protect app user identity		
Originator: Application Development Team		

Fit Criterion: Given that customer and client engage in an exchange of information, client and customer can only see the contents of the chat, the optional uploaded receipt and the order number	
Supporting Materials: Business Use Cases 1,2,3,4 Definitions and assumptions	**Conflicts:**

Requirement ID: FK13	**Requirement Type:** Service level agreements	**PUC:** Making Payment
Description: Vendor of Built-in payment option must provide appropriate solutions and be available for consultations in cases of integration problems		
Rationale: Smooth integration of technology into the app matters for an excellent customer experience. If developers are having difficulty maintaining certain features due to lack of aid from the vendor, it is enough to switch to other vendors as customer service levels will be impacted eventually.		
Originator: Developer		
Fit Criterion: Given that a certain third party vendor is selected, developers can easily find FAQ pages with regards to feature integration as well as the customer service number for developers.		
Supporting Materials: Square developers guide Braintree developers guide Paypal developers guide	**Conflicts:**	

Requirement ID: FK14	**Requirement Type:** Privacy	**PUC:** Making Payment
Description: Customer transaction history does not contain payment credentials. Only authentication/ confirmation codes are stored per transaction and are tagged to order number		
Rationale: To protect customer financial information in the case of security breaches		
Originator: App Owner		
Fit Criterion: Given that clients and customers are on their order history pages, they can only view the order number which acts as a link to show them the transaction history saved in the cloud. Only chat history, optionally uploaded receipt, payment method and amount paid are shown.		
Supporting Materials: Google cloud database API Definitions and assumptions	**Conflicts:**	

Requirement ID: FK15	**Requirement Type:** compatibility	**PUC:** Making Payment
Description: The application must be adaptable to changes in payment options or features contained in the APIs		
Rationale: to ensure a smooth experience for customers every time		
Originator: Customer		

Fit Criterion: Given that Square API or Tbucks API experiences an upgrade, application must be able to comply with new systems requirements within 5 hours of the new upgrade	
Supporting Materials: Definitions and Assumptions Square checkout integration guide Transactcampus Documentation	**Conflicts:** FK11, FK 19

Requirement ID: FK16	**Requirement Type:** reliability	**PUC:** Making Payment
Description: Due to use of external vendors for payment options, any messages about system upgrades or system unavailability shall be communicated to the customer within 3 minutes of the reception of the notice		
Rationale: To ensure that customers are able to pay using their selected method of choice and are not struggling to make a payment to clients		
Originator: App User		
Fit Criterion: Given that the development team receives notice of major upgrades from vendors, a system wide notification goes out to all app users informing them of any service disruptions		
Supporting Materials: Strategic Dependency I* Definitions and Assumptions Onion Model Square checkout integration guide Transactcampus Documentation	**Conflicts:** FK15	

Requirement ID: FK17	**Requirement Type:** Availability	**PUC:** Making Payment
Description: The payment methods section should always be present in the user profile page of the app and allow customers to make changes to their credentials whenever they like.		
Rationale: If a customer's credit card number changes, they should have the option to upgrade it before engaging with a client in a transaction		
Originator: Customer		
Fit Criterion: Given that the customer is on their user profile page, customer has the option to edit method of payment acceptance as well as payment credentials		
Supporting Materials: Context Diagram Definitions and Assumptions	**Conflicts:** FK15, FK16	

Requirement ID: FK18	**Requirement Type:** extensibility	**PUC:** Making Payment

Description: As a third-party app, the app should be flexible enough to incorporate other modes of payment in case there is an increasing demand on university campuses outside of UofT	
Rationale: We don't want to limit the scope of the app to only uoft campuses in the future. Other university campuses may have their own version of t-bucks.	
Originator: App Owner	
Fit Criterion: Given that the app owner is approached by administrative heads of other universities to support their on-campus smart cards, the development team can add new forms of payment to the app	
Supporting Materials: Definitions and Assumptions Context Diagram Onion Model	**Conflicts:**

Requirement ID: FK19	**Requirement Type:** Performance Efficiency	**PUC:** Making Payment
Description: Any and all notifications about payment or payment options will appear on the app interface within 2 seconds at least 90% of the time		
Rationale: Issues around payment can be a major cause of conflict between app users. Semi-instantaneous messaging can help mitigate potential conflict, especially in scenarios were the client is at the counter buying the food for the customer		
Originator: App User		
Fit Criterion: Given that the app uses external vendors, the developers shall run a daily scan of their systems to ensure all components are working in sync. Any anomalies in the system shall be communicated to the app users within 10 minutes of detection		
Supporting Materials: Definitions and Assumptions Context Diagram Onion Model	**Conflicts:**	

Requirement ID: FK20	**Requirement Type:** maintainability	**PUC:** Making Payment
Description: The transaction information obeys all provincial and federal laws when considering the transfer of large amounts of money into client account over a short period of time		
Rationale: Although clients can make pocket change through this application through tips and gratuities, we need to ensure federal income tax laws are not violated through this process		
Originator: Application Operations Team		
Fit Criterion: Given that the clients are frequently receiving tips, the system must keep track of all client tips in order to send accurate income tax papers		
Supporting Materials: Strategic Dependency I* CRA tips and gratuities webpage Definitions and Assumptions	**Conflicts:**	

Context Diagram	

11 ARCHITECTURAL DESIGN

This section represents the stage you presented in week 12. Use comments and feedback received to improve aspects of your architectural design if applicable.

11.1 KEY DRIVERS AND ARCHITECTURALLY SIGNIFICANT REQUIREMENTS

Features	Architecturally Significant Requirement	Stakeholder	Use Cases
User Account Management	Privacy & Security, Availability, Login and Validation, Compatibility, Data Storage and Retrieval	App User, Developers, Operations Team	PUC1, PUC5,
Location and Map	Accuracy and Precision, Data Storage and Retrieval, Performance, availability, Tracking App User, Compatibility, Messaging and Notification,	App User, Developers	PUC2, PUC3, PUC4
Information Exchange	Data Storage and Retrieval, Performance, Availability, Tracking App User, Compatibility, Messaging and Notification	App User, Operations Team	PUC1, PUC2, PUC3, PUC4, PUC5
Payment	Privacy & Security, Payment, Availability, Compatibility, Messaging and Notification, Data Storage and Retrieval	App User, Operations Team	PUC5

11.1.1 Functional Drivers

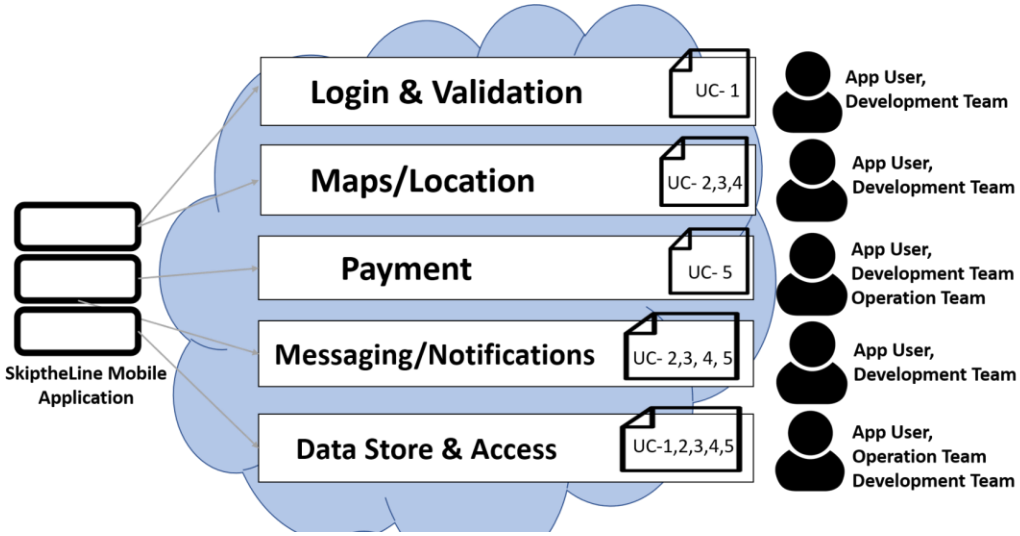


Figure 8: Functional Drivers of SkiptheLine

11.1.2 Architecturally Significant Requirements

Functional Requirement 1: Validation and Authentication

Requirement ID: *CAA02*	**Requirement Type:** Functional	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure secured authentication and validation of User credentials through email or multi-factor authentication.		
Rationale: To ensure that there is a secured validation and authentication of User information within the SkiptheLine database.		
Originator: System Administrators		
Fit Criterion: The system shall require new Users to swipe their fingerprint three times and must capture and store the fingerprint automatically in SkiptheLine system within 2 seconds after the last swipe. Each new User shall receive a verification email message and the system shall automatically generate the date and time of when the email was released. The system shall give the User the option to choose either biometric or email verification.		
Supporting Materials: A2 Vision: BUC #1, PUC Table	**Conflicts:** TBD	

Functional Requirement 2: Tracking App User

Requirement ID: ZL01	**Requirement Type:** Functional	**PUC:** Search Client
Description: The app shall allow Users (Customer and Client) to view the eateries nearby that are open and available.		
Rationale: Need to let Users know the available options so that they can make decisions on what to eat.		

Originator: Customer and Client	
Fit Criterion: When a User initiates a search, the app shall: · provide a full list of eateries that are open on campus, based on current time and location (will be referred to as "search result list page"); · by default, the list shall be sorted ascendingly according to the distance between each eatery and the Customer's location.	
Supporting Materials: A2 Vision	**Conflicts:** TBD

Functional Requirement 3: Tracking App User

Requirement ID: ZL05	**Requirement Type:** Functional	**PUC:** Search Client
Description: The app shall allow app Users to be identified as Clients.		
Rationale: To encourage app Users, and even those who originally want to use the app as Customers, to become Clients and help others.		
Originator: Client		
Fit Criterion: On the eatery's information page, there shall be a button in the available Client list to let the app User sign up as a Client for this eatery.		
Supporting Materials: A2 Vision	**Conflicts:** TBD	

Functional Requirement 4: Messaging and Notification System

Requirement ID: *RA03*	**Requirement Type:** Functional	**PUC:** Place Food Order
Description: SkiptheLine app shall integrate Instant Messenger and Notification System. App Users can chat with one another regarding: Food Orders, Payment methods, Order Confirmation and track one another. The app will notify App Users of any notifications (new messages).		
Rationale: In order to fasten the process of placing food order, Customers will directly message Clients.		
Originator: Development Team		
Fit Criterion: Given Customer finds Client(s) going to their desired Eatery through the Instant Messenger, Customer can message Client food order details and payment method and within 30 seconds receive food order confirmation.		
Supporting Materials: A2 Vision: Strategic Dependency ModelBusiness Use Cases: 3,4,5	**Conflicts:** TBD	

Functional Requirement 5: Tracking App User

Requirement ID: DZ03	**Requirement Type:** Functional	**PUC:** Search for Customer
Description: The App shall enable a GPS tracking network to update the location of the Client and Customer. This shall enable the Customer to search for available Clients around them, which shall return a list of available Clients based on their locations. After the Client has purchased the order for the Customer, they shall also be able to locate the Customer on the map and vice versa.		
Rationale: The Client and the Customer should be able to locate each other on the virtual map on the		

app.
Originator: Client and Customer
Fit Criterion: Once the Client has purchased the order, the Client and the Customer should be able to see each other's live location on the virtual map on the App. The location shall be updated instantly.

Supporting Materials: A2 Vision Business Uses Case: 4	**Conflicts:** TBD

Functional Requirement 6: Data Storage and Retrieval

Requirement ID: FK05	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The application shall contain an 'order complete' button for the Client to press once he has received his payment. The records will show 'payment pending' until the Client clicks on the 'order complete' button. An order number shall be produced and the order number shall link to a database to allow Client and Customer to view chat history. Nothing except the order number is saved on the app User's phone.		
Rationale: to be able to log the end of a transaction in the database, end the transaction in the app and store transaction information in the cloud database		
Originator: App Developer		
Fit Criterion: When Client initially accepts order from Customer, the "complete order" button pops up at the top of the app screen within 5 seconds of order confirmation. Transaction details should be uploaded into the cloud within 5 seconds of the press of the button and should be cached in the cloud until batch uploads are pushed to the database every 12 hours.		
Supporting Materials: Business cases 2,3, Definitions and assumptions, Context Diagram	**Conflicts:** TBD	

Functional Requirement 7: Payment

Requirement ID: FK04	**Requirement Type:** Functional	**PUC:** Making Payment
Description: The Customer 'order in progress' page shall contain a 'Make Payment' button which takes the Customer to a new page on the app containing fields such as "order amount", "tips and gratuities", payment options and a "submit payment" button. The tips and gratuities section will have tip denominations from 5%, 10% , 15% of the bill or Customers will have the option to manually type in their desired amount. If the in-app payment option of using Square is selected, the Customer will be taken to an external page via an API to continue the transactions. If cash, tbucks or eTransfer options are selected, a 'submit payment' option occurs and Customers can press it if the payment has been made.		
Rationale: Since the food is not being ordered via the app, Customers will have to manually insert the amount from their order into the 'order amount' field and manually select any tips or gratuities options. Since Cash and eTransfer options occur outside of the app, Customers just click 'submit payment' to end transaction		
Originator: Application Operations Team		

Fit Criterion: Given that the Customer has an order in progress and the Client arrives with the Customer food, Customer clicks on the 'Make Payment' page and fills in all the appropriate. information	
Supporting Materials: Definitions and assumptions, Context Diagram, Business use case 1,2,3, PayPal checkout integration guide, Transact Campus Documentation	**Conflicts:** TBD

Non-Functional Requirement 8: Performance

Requirement ID: *CAA11*	**Requirement Type:** Performance	**PUC:** Create a SkiptheLine profile
Description: The system shall be robust and not crash due to overload of App User Data or simultaneous login or registration.		
Rationale: To ensure that users are able to login or create a profile at any time and during peak periods of the application.		
Originator: Developers		
Fit Criterion: There must be 95% efficient login or registration of App User during peak periods of the application. The application shall generate a report back to system administrators within 2 seconds in the event of a crash.		
Supporting Materials: BUC #1, PUC Table, SD Model (A2 Vision).	**Conflicts:** TBD	

Non-Functional Requirement 9: Privacy

Requirement ID: *CAA15*	**Requirement Type:** Privacy	**PUC:** Create a SkiptheLine profile
Description: The system shall ensure that existing users are actively notified of changes to the SkiptheLine application's Terms of Use and Privacy Policy.		
Rationale: To ensure that users feel informed and comfortable in their use of SkiptheLine and how their data will be used.		
Originator: Legal team		
Fit Criterion: The legal team shall create a statement for distribution 21 days prior to any changes being made. Users shall be presented with a notification on the application homepage regarding these changes and be sent an email 14 days prior to changes coming into effect.		
Supporting Materials: BUC #1,2,3,4,5 PUC Table, Onion Diagram (A2 Vision).	**Conflicts:** TBD	

Non-Functional Requirement 10: Availability

Requirement ID: *RA14*	**Requirement Type:** Availability	**PUC:** Place Food Order

Description: SkiptheLine app is available 23 hours a day, 7 days a week and 365 days per year; as SkiptheLine purpose is to enhance food consumption experience better. However, most Eatery Restaurants on campus are open on certain hours, the app will display alternatives on campus (i.e. Vending Machines nearby). The app may have scheduled downtime or for technical reasons which is why the expected available hours for the app is 23 hours a day.	
Rationale: Since students have different food consumption timing (i.e. exam time, hectic schedules, overnight shifts) and are always looking for food options; SkiptheLine app tries to be available 24/7.	
Originator: Operation Team	
Fit Criterion: During exam and midterms, SkiptheLine app will be available 24 hours a day, 7 days a week; running 99% of the time without major technical issues.	
Supporting Materials: A1 & A2 Vision Business Use Cases: 1,2,3,4,5	**Conflicts:** TBD

11.1.3 Key Quality Scenarios

Quality Scenario 1: Performance

Overview	How the system performs when there is overload or simultaneous login or registration of many Users.
System State	Application performance/User login or registration is working quickly, initially as expected.
Environment changes	The need to maintain performance when there is an overload of app usage due to simultaneous login and registration during peak period.
Required System Behavior	The application must be able to maintain a 95% efficient login/registration process during peak period.

Quality Scenario 2: Availability

Overview	How the system provides 24/7 login/registration services and handles downtime or component failure
System State	The app system is operational 24/7 so that Users can create their profile and log in at any time.
Environment Changes	The system becomes slow or crushes during User profile creation or login due to component failure.
Required System Behaviour	The system should generate a report back to system administrators with component stack traces in the event of node or component failure and restore availability within 2 hours.

Quality Scenario 3: Privacy and Security

Overview	How the system ensures privacy of Clients, Customers and their information
System State	The app communicates with the server only when the “Place order” button, “make

	payment" button and "order complete button" is pressed. All app Users see other app Users on the map but cannot communicate with each other unless an order is placed between the two.
Environment Changes	Customer chooses a Client under the pretext of wanting to place an order but tries to communicate about non-order related topics.
Required System Behaviour	The Client can click on "cancel order" and within 30 seconds report the issue to the Operations Team. System will cancel the order within 1 minute and flag specific transactions as needing immediate investigation to the Operations Team. If the investigation (done within 48 hours of the initial report) concludes improper conduct, Customer rating will be penalized and upon three misconduct reports, the system will ban the Customer's account.

11.2 CONTEXT VIEW

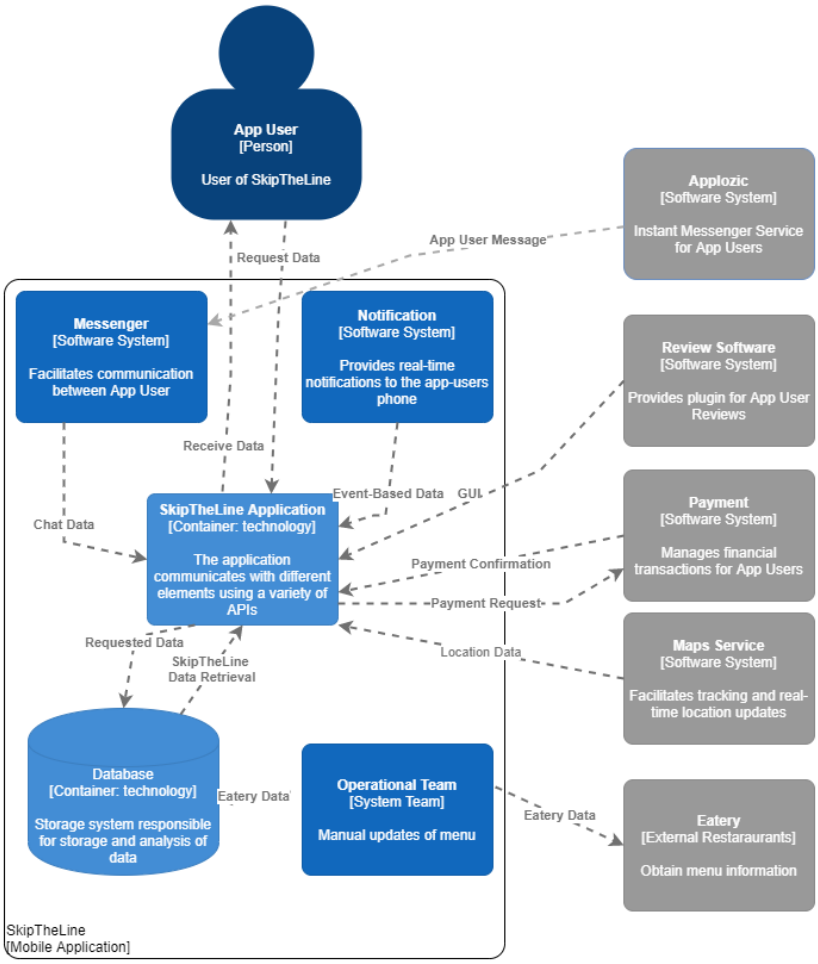


Figure 9: Context View Diagram

The App User is dependent on the SkiptheLine App for placing an order, searching for a Client, messaging a Client and to a certain extent, paying for their food. What is excluded on this diagram is that the Customer had the option to use external peer-to-peer payment methods to pay for their food but at their own discretion. This selection is recorded in our database.

The instant messenger allows Customers and Clients to chat about their order preferences, any delays that may be occurring (e.g if eatery is out of a specific food item) and lets all app Users to upload any pictures related to their transaction such as a receipt or a price change in the menu.

The notification system will allow App Users to receive push notifications such as when a Customer requests to connect with a Client to place an order or when a Client confirms that they have been reimbursed by the Customer for their order.

The database contains records of all App User's interactions including chat history, pictures uploaded to the instant messenger and the transaction history which will include the total order amount and the payment method employed.
The Operations Team manually updates the menu and their associated prices visible on the app, the only components in the SkiptheLine system that will communicate with the campus eateries for that information.

Elements shown in the grey constitutes external components of the app. The review software is an external User interface plugin that will be integrated into the app to allow Clients and Customers to rate each other's service and professionalism.

The payment system is a third-party product whose User interface will be built into our app but the payment processing will occur on a third-party server using a compatible API software.

The Maps system will allow Customers to track the Clients heading towards their desired eatery and will allow both Customer and Clients to locate each other in order to exchange goods for payment.

11.3 KEY ARCHITECTURAL DESIGN DECISIONS

11.3.1 RA & Rational

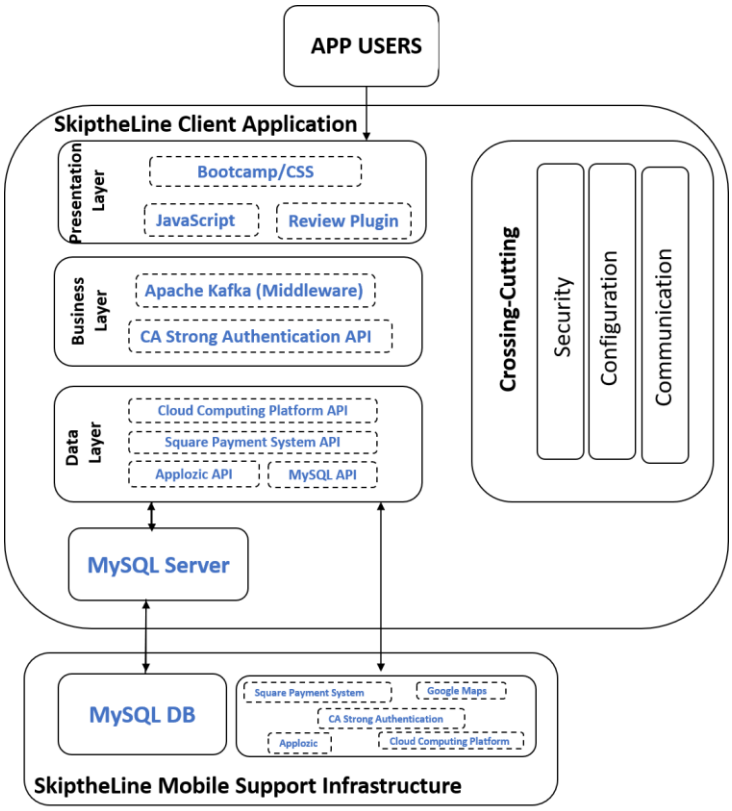


Figure 10: Reference Architecture

As a team, we attempted to play the smart decisions game but discovered that none of the Reference Architectures included in the red cards fully satisfies the capabilities of our application; some included functionality that was beyond the vision of our SkiptheLine Application. For our purpose, we found the Mobile Application Archetype with a Thin Client to be most suitable as a Reference Architecture (Microsoft, 2020).

To fulfil our Architecturally Significant Requirement of Compatibility, we decided to build our application as a Thin Client. Not all cell phones have the hardware capabilities to process and store large amounts of data. To ensure our app is compatible with several devices as possible while providing similar levels of performance, building it as a thin Client was the best choice. This means, however, that internet connection cannot be undermined as far as Customers locating willing Clients is concerned (Microsoft, 2020). The App Users interact with the application interface from their mobile phone but the login and validation, data access and retrieval and location and tracking services occur via APIs and servers.

The Diagram above shows the three different layers of our SkiptheLine Application and the elements required for its optimal functioning; Presentation Layer, Business Layer and Data Layer. The Presentation Layer contains elements that provide the User interface. Our Language of choice for the application is Javascript, the common language supported by many products available on the market. The business layer provides the logic involved with retrieval, processing, transformation and management of

application data; thus, the layer implements middleware. The Business Layer contains Apache Kafka and CA Strong Authentication User Account Management software. Based on the blue cards from the Smart decisions game, Apache Kafka was identified as best software handling real time data. This is necessary for real time tracking of Clients and Customers who are engaged in an order. The components in the data layer help provide access to the data that is created within the system as well as data coming in from other networks. This is the layer where the cloud server supporting our app lies as well as the APIs which provide communication to third-party services to our system. The MySQL server and the other products listed in the support infrastructure are the third-party vendors whose products we access through our data layer to achieve full functionality of our app.

Although the Lambda and Extended Non-relational Architectures were the two Reference Architectures, from the smart decisions game, which fit better than the others, here are a few reasons why we decided not to select either. **Please see the appendix VI for a more in-depth analysis.**

Lambda	Extended Non-Relational
1. Lambda is for massive quantities of data, but for now we only focus on an estimation of 50,000 MAU on U of T campus, and that does not qualify as big data; 2. Lambda is good for real-time big data processing and analysis, but our app does not have that requirement;	1. Extended Relational doesn't provide real-time analysis and historical data simultaneously. App Users need to be able to simultaneously view Eatery Restaurants information and find near-by Clients. 2. Extended Relational depends on in-memory techniques for scalability. However, SkiptheLine App needs real-time data being stored and accessed for accuracy in location, finding Clients, notifications, messages.

11.3.2 Discuss and Explain Design Decisions

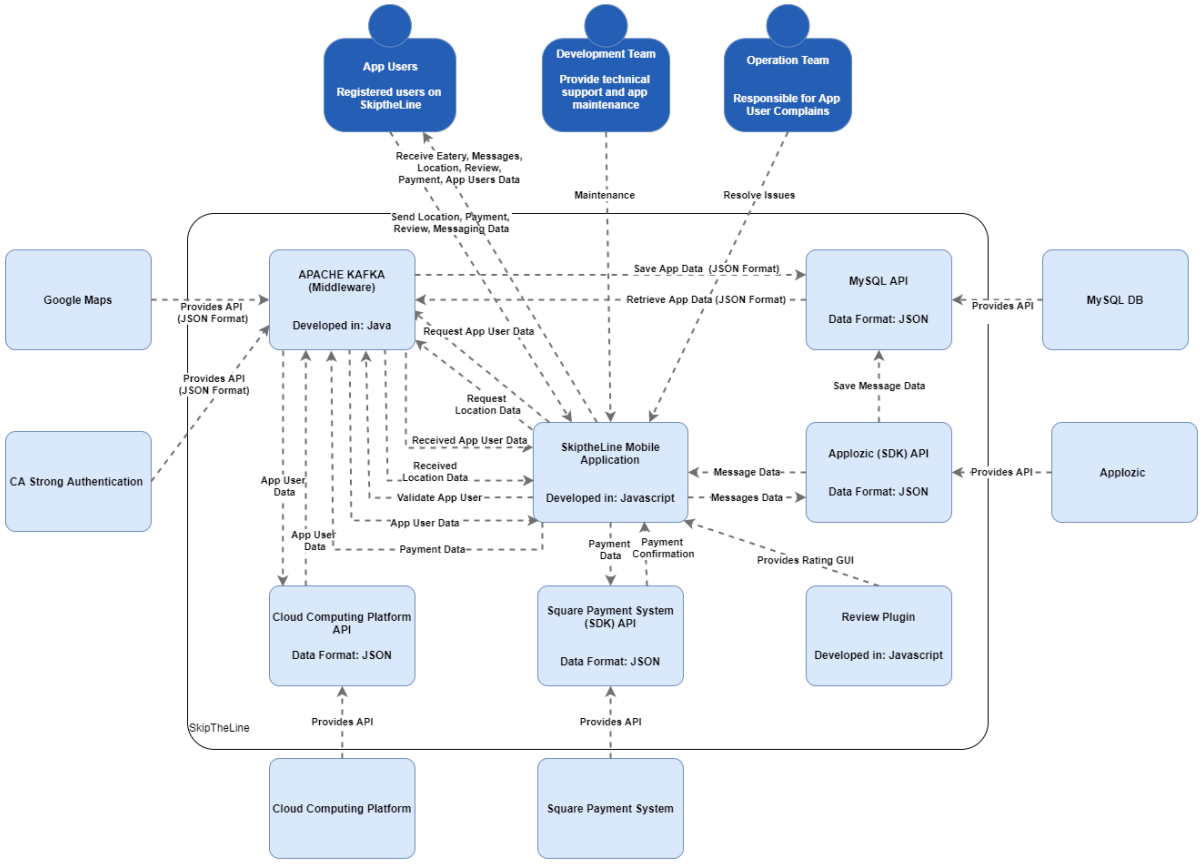


Figure 11: Functional View of SkiptheLine

As previously mentioned, we attempted to play the Smart Decisions game but soon found that none of the reference architectures fit into the design of our product. We used the red and blue cards as reference to become familiar with the different technology families and the products on the market. The following products were selected, for each of our elements, through rigorous research of numerous products that are available on the market. In this section we talk about the selected products and the reasoning behind it. Please refer to the appendices I-IV for full product evaluations.

Due to the Thin Client nature of our app, it is hosted on a cloud server where all the information processing takes place. Apache Kafka is supported by numerous cloud servers so a specific one was not selected. Information collected from the User is inputted and processed in the cloud via Apache Kafka which separates, organizes and prepares data to be written into the MySQL database. Kafka and MySQL integrate through the Debezium platform (not shown) and communicate through the MySQL API. The data travelling between elements is JSON formatted for all APIs and Servers in our SkiptheLine Application.

User Account Management (Functional Requirement: Login and Validation)
The decision to choose **CA Strong authentication** was influenced by the following priorities:

Cost-Efficiency: CA Strong authentication requires a seemingly lower initial capital investment and offers long term cost efficiency.
Security: CA Strong provides high security to support Users' accounts and payment information. It offers a variety of authentication methods, including passwords, SMS, email, knowledge-based authentication, two-factor software tokens, and hardware credentials. This helps eliminate the risk of stolen credentials because it never stores credentials thereby making them unbreachable.
Integration compatibility: It is compatible with several devices, applications and integration options, such as integration with SAML, APIs, and Remote Authentication Dial-In User Service.
Scalability: It is much easier to scale up without impacting the overall performance of the application. And offers flexibility for Users to select from varied methods of logins and validation

Instant Messaging (Functional Requirement: Messaging and Notification system):
Choose **Applozic** as instant messaging service provider after considering:
Cost Efficiency: Since nearly all third-party SaaS design their pricing plan based on the number of active Users, we use 50,000 as an estimation of our target MAU (approximately 50% of currently enrolled students plus 30% of faculty, staff members and librarians). Applozic is relatively inexpensive to serve 50,000 MAU.
User Management: Applozic has features like ban (malicious) Users and profanity filter, which fits in our requirement for instant messaging systems.
Support: Applozic provides SLA-based support and dedicated account manager, while other products and plans with similar pricing can only provide limited support on business days.
Message retrieval & data export: Lifetime message history retention is promised but export options weren't specified. Still it wins over other competitors as they either do not mention message retrieval policy or charge a premium for the service.
Note: The qualities like privacy, security and scalability, the desirable features like media sharing are usually provided by default by instant messaging SaaS; only those features and services that are different across the solutions are compared in the evaluation. See Appendix II for details.

MySQL (Functional Requirement: Data Storage and Access):
The process in selecting **MySQL DB** was influenced by the following priorities:
Cost Efficiency: After examining the alternatives (see Appendix III) for Data Storage and Access, MySQL best suited SkiptheLine in terms of least amount of economical investment.
Real-time Data Manipulation: SkiptheLine requires data to be filtered, computed, stored and retrieved per second. MySQL can manipulate and generate data quickly providing data that is accurate and precise.
Scalability and Extensibility: MySQL is able to migrate (newer version) and grow (more space) its storage. As the App Users data increases, SkiptheLine may need to migrate or grow their Database.
Data maintenance: MySQL can effectively perform CRUD operations (Insert, Update, Delete and Retrieve Data) for large amounts of data.
Performance & Speed: MySQL is fast and takes seconds to modify and retrieve data. SkiptheLine benefits from MySQL as it processes data fast.

Map/location (Functional Requirement: Tracking App User):
The decision to choose **Google Maps SDK** is supported by follow rationals:
Maintainability: The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures.

Location tracking: The Geolocation API returns the precise location of a User's device based on Wi-Fi or cell towers. The request will return a JSON-formatted response defining a location and radius.
Customization: Markers indicating User's locations can be added to the map with the GoogleMap.addMarker(markerOptions) method. Markers are designed to be interactive. Setting a marker's draggable property to true can allow the User to change the position of the marker.
Accuracy: The immense data in Google Maps' database means it can outperform Mapbox and OSM when Users are zooming closely into the map.

Payment Technology of Choice: Square (Functional requirement: Payment)
Due to the design of our App and the availability of Peer to Peer payment methods, we assume that the majority of our users will opt out of using the in-built payment method. All UofT students are able to accept credit card payments in the form of t-bucks deposits, which is instant as opposed to having to wait 1-2 weeks for the payment to be sent from our end to the App User's bank Account. Nonetheless, we have decided to include this option so as not to inconvenience our app users
Convenience: Square as the option to allow users to key in their credit card numbers as opposed to requiring users to log into an account
Cost Efficiency: Square is not the most cost effective and the rate per transaction is slightly higher at 3.4%+0.15. However, there are no monthly fees and this is important for SkiptheLine App as we assume the delay in receiving payment will encourage app users to use Peer to Peer payment methods such as t-bucks, e-transfer or cash.
Security: All payments are encrypted end to end. From the backend perspective, they have a fraud monitoring feature which uses machine learning to pick up on any unusual patterns.
Ease of Integration: Customizable User Interface to allow payment to occur within the App as opposed to a payment portal without extra fees
Speed: Speed of transaction is instant and payment is available on SkiptheLine Business Account within 1-2 days. This is faster than the competitors and allows us to send money to our App Users faster.

11.3.3 Deployment View

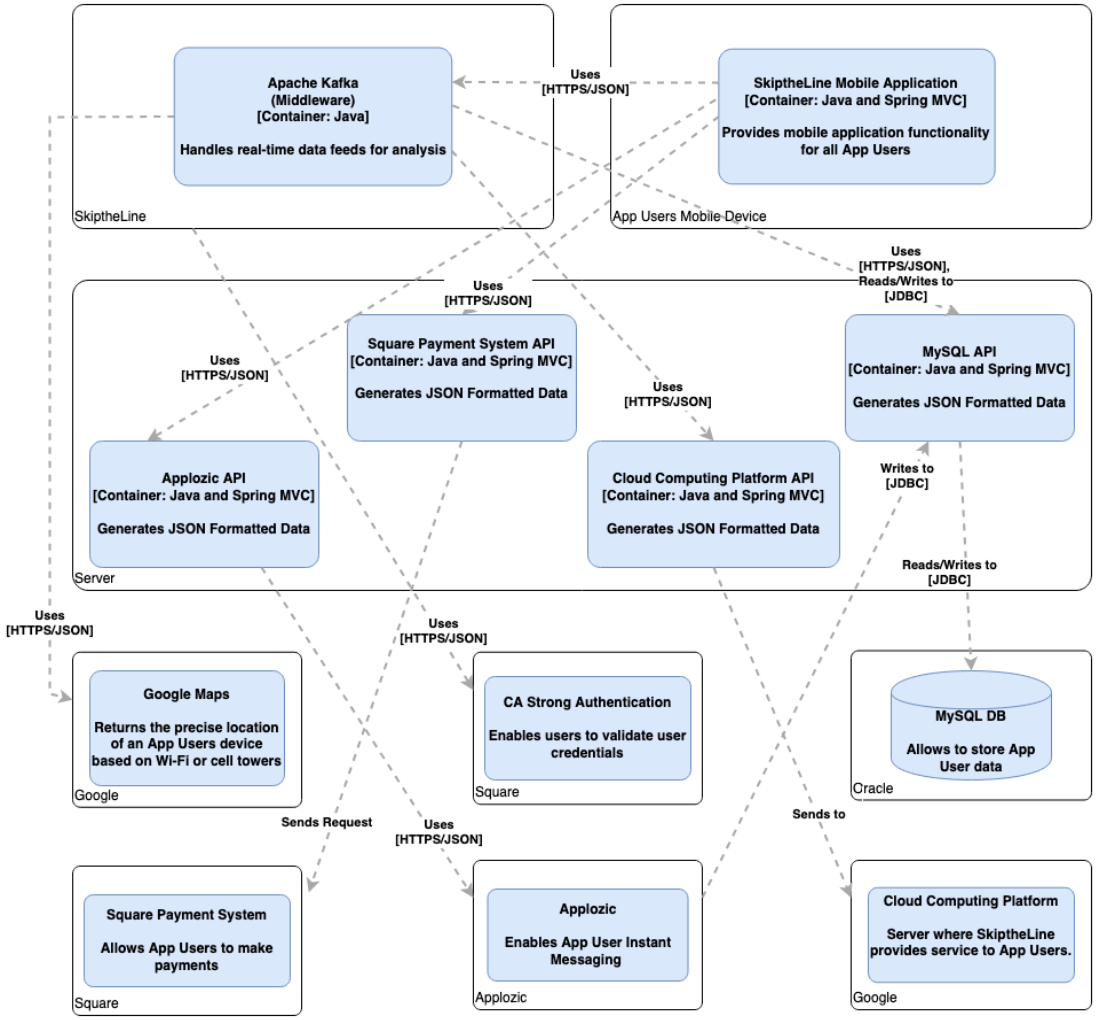


Figure 12: Deployment View of SkiptheLine

The deployment view diagram represents the model's hardware and software elements and the data communication paths between them. App User data and information is processed between the MySQL database and the Cloud server which hosts APIs of Applozic (messaging), Square Payment System, Cloud Computing Platform, and MySQL, except for Google Maps and CA Strong Authentication which directly send to and receive data from Apache Kafka through APIs. These APIs generate data in JSON format and send requests to their external service provider's servers and receive response data, which is then stream processed by Apache Kafka before communicating directly to App Users' mobile applications.

12 OPEN ISSUES

	Key driver	How addressed
Quality	**Privacy and security:** The system shall protect the privacy of User personal information, credentials and payment information.	Fully addressed: CA Strong authentication provides maximum security to prevent unauthorised access to User information. Multi-factor authentication can also protect User payment information using hashing or masking.
	Accuracy and precision: The system shall enable accurate location tracking and display on the virtual map.	Fully addressed: Google Maps Geolocation API returns the precise location of a device based on Wi-Fi or cell towers, which performs well even inside buildings where GPS signals are often obstructed by surrounding buildings and the crowd.
	Performance: The system shall be robust and not crash due to overload or simultaneous login or registration of Users.	Fully addressed: Our app has Apache Kafka with a rich pre-built ecosystem that organises and manages data in a reliable and high performance mode. It also has MySQL which is fast and takes seconds to modify and retrieve data.
	Availability: The system shall be available 24/7/365, and User shall be able to login or register at any time, and any downtime shall not exceed 2 hours in the event of node or component failure.	Partially addressed: To the best of our knowledge, all components and patterns support redundancy. The exact deployment configuration is yet to be employed and we have not simulated or tested this. Q. How do we set up replication and clustering in each of the components? Next step: Create a test environment and wrote a code that simulates failures
	Compatibility: The system shall be compatible with mobile devices, applications and current operating systems.	Partially addressed: To the best of our knowledge, all components, reference architectures and technologies will be compatible with most devices, applications and operating systems. The exact deployment configuration is yet to be done. Q. How do we set up testing for compatibility? Next step: Create a test environment to pilot compatibility for devices, applications and operating systems.

Functional	**Validation and authentication**: The system shall ensure secured authentication and validation of User credentials through email or multi-factor authentication.	Fully addressed: We have incorporated CA Strong authentication provides high security to support Users' accounts and payment information. It offers a variety of authentication methods, including passwords, SMS, email, knowledge-based authentication, two-factor software tokens, and hardware credentials. This helps eliminate the risk of stolen credentials because it never stores credentials thereby making them unbreachable.
	Payment: The Customer 'order in progress' page shall contain a 'Make Payment' button which takes the Customer to a new page on the app containing fields such as "order amount", "tips and gratuities", payment options and a "submit payment" button. If the in-app payment option of using Square is selected, the Customer will be taken to an external page via an API to continue the transactions. If cash, tbucks or eTransfer options are selected, a 'submit payment' option occurs and Customers can press it if the payment has been made.	Partially Addressed: Our app is different from the traditional mobile food delivery apps as we facilitate a peer to peer service. As such, the payment options our Users have access to are also peer to peer. To ensure our Users have the flexibility to choose from an already limited amount of options, we include the top 4 payment options in our app. Cash, e-transfer, tbucks, and an integrated payment option using square. The integrated payment method using Square is the most secure and convenient for the Customer but it is slow for the Client as the Client will receive their payment after SkiptheLine Operations Team approves the fund transfer to the Client's bank account.
	Tracking app Users: The App shall enable a GPS tracking network to update the location of the Client and Customer. This shall enable the Customer to search for available Clients around them, which shall return a list of available Clients based on their locations. After the Client has purchased the order for the Customer, they shall also be able to locate the Customer on the map and vice versa.	Fully addressed: The system has incorporated Google SDK including Google Places API, which complements GPS tracking system with cellular locating technology to return the precise location of a User's device. The system also enables the User to change their location on the map manually in case the location detected by the system isn't accurate enough.
	Messaging and Notification: SkiptheLine app shall integrate Instant Messenger and Notification System. App Users can chat	Fully addressed: Applozic is a messaging and notification system in our app purposely for providing SLA-based support and dedicated

	with one another regarding: Food Orders, Payment methods, Order Confirmation and track one another. The app will notify App Users of any notifications (new messages).	account manager.It ensures that Lifetime message history retention is promised.
	Data storage and retrieval: The application shall contain an 'order complete' button for the Client to press once he has received his payment. The records will show 'payment pending' until the Client clicks on the 'order complete' button. An order number shall be produced and the order number shall link to a database to allow Client and Customer to view chat history. Nothing except the order number is saved on the app User's phone.	Fully address: Our app has employed MySQL that filters, computes, stores and retrieves data per second. MySQL manipulates and generates data quickly providing data that is accurate and precise.

13 CHANGES MADE IN THE COURSE OF ANALYSIS

Based on the feedback received from Assignments A2-A4, we have made the following changes to our system

We added to the purpose of our application and explained what problem our app addresses that is not currently being addressed. We also broke down our goals and mentioned what the goal is addressing and how to measure it.

We decided to build a multi-factor authentication feature in the log-in function to better protect users' account privacy and security. The feature includes the option of biometric authentication (i.e. fingerprint) but also allows email/SMS double-validation. This influenced our decision to choose CA Strong Authentication to support the user account management of the app, because it supports a variety of authentication methods.

We also decided to choose Applozic as the instant messaging service provider to support the messaging and notification function of the app, instead of building our own messenger. Our rationales include the fact that Applozic is relatively low-cost given our estimated number of users, its capacity to provide profanity filter and banning/blocking users, and sufficient SLA support it provides.

Initially, we planned all payment of food items to happen outside the application as the app was mainly going to be responsible for facilitating communication of food orders, not for placing orders with specific eateries. After receiving instructor feedback, we decided to include an in-app payment system that gives users the option to pay using their credit card from within the app. We are still allowing customers to pay using Peer to Peer payment methods like cash, e-Transfer, and T-Bucks, at their own discretion. We are assuming that these Peer to Peer payment options will be more popular than the built-in system as clients will receive the money within minutes versus weeks. The money from the built-in payment option will be deposited into the SkiptheLine business account which will then be processed and sent to the client's bank account within 1-2 weeks of payment reception. Functional Requirements FK04, FK08, FK09 have been changed to include an in-app payment system. Non-Functional Requirement FK13 has been completely changed since A3. Some conflicts were found and added to the payment nonfunctional requirements.

In the Product Boundary Section, we represented the Client and Customer as primary stakeholders of the SkiptheLine system in the Use Case diagram, in contrast to the Eatery which has no direct interactions with the SkiptheLine app. We clarified the product boundary by representing the Client and Customer on the left side of the SkiptheLine system as primary users and the Eatery on the right side as a non-user. We also displayed the interactions between the Client and the Eatery outside the SkiptheLine system to suggest that the part of the process should be done by users off-line.

As we included in our A4 but did not highlight appropriately, all the commercially available products we researched for our individual requirements and the criteria we judged them by are **available in the Appendix (Please see below, after references).**

REFERENCES

Apache Kafka. (2017). Introduction: Apache Kafka is a distributed streaming platform, What exactly does that mean?. Retrieved from https://kafka.apache.org/intro#:~:text=The%20streams%20API%20builds%20on,among%20the%20stream%20processor%20instances.

AWS.(2020). Amazon Managed Streaming for Apache Kafka (Amazon MSK). Retrieved from https://aws.amazon.com/msk/

Big Data - Blue Cards. (2020). Retrieved from https://smartdecisionsgame.com/

Braintree.(2020). Data Security. Retrieved from https://www.braintreepayments.com/ca/features/data-security#:~:text=Cardholder%20data%20is%20managed%20in,connected%20to%20via%20the%20internet.

Brosh, T.A. (2019, September 1). How to choose the right database. Medium. Retrieved from https://towardsdatascience.com/how-to-choose-the-right-database-afcf95541741

Cervantes, H., Haziyev, S., Hrytsay, O & Kazman, R. (April, 2015). Smart Decisions: An Architecture Design Game. Retrieved from https://resources.sei.cmu.edu/asset_files/Presentation/2015_017_101_438650.pdf

Confluent Cloud (2020). Fully Managed Apache Kafka® As a Service on Azure. Retrieved from https://www.confluent.io/confluent-cloud

Confluent Cloud.(2020). Fully Managed Apache Kafka® As a Service for GCP. Retrieved from https://www.confluent.io/gcp/

Dimi. (2020, January 23). Cloud Server Hosting vs Traditional Hosting (Full Guide) |. Retrieved from https://themefuse.com/difference-between-cloud-server-and-normal-server/#:~:text=Normal%20servers%2C%20in%20fact%2C%20refers,operations%2C%20and%20allow%20enough%20customization

Finances Online.(2020). Braintree Review. Retrieved from https://reviews.financesonline.com/p/braintree/#:~:text=Braintree%20Pricing%20Plans%3A&text=2.9%25%20%2B%20%240.30%20charge%20per%20transaction,fee%20is%20assessed%20for%20chargebacks

Firebase Authentication | Simple, free multi-platform sign-in. Retrieved 9 June 2020, from https://firebase.google.com/products/auth

Firebase Products. Retrieved 9 June 2020, from https://firebase.google.com/products

Fit Small Business. (June 4, 2020). Square vs Paypal: Price, Features & What's best in 2020. Retrieved from https://fitsmallbusiness.com/square-vs-paypal/

Gartner. (2019). Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018. Retrieved on April 7, 2019 from https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-system/

Google Maps Platform.(2020 June 5). Geolocation Developer Guide. Retrieved from https://developers.google.com/maps/documentation/geolocation/intro

Google Maps Platform.(2020 June 5). Maps SDK for Android Developer Guide. Retrieved from https://developers.google.com/maps/documentation/android-sdk/intro

Lambda architecture. (2020, February 28). Retrieved from https://en.wikipedia.org/wiki/Lambda_architecture

Microsoft.(Jan 13, 2010). Chapter 24: Designing Mobile Applications. Retrieved from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658108%28v%3dpandp.10%29

Mindfire Solutions. (2017, April 10). A comparison between MySQL vs. MS SQL server. Medium. Retrieved from https://medium.com/@mindfiresolutions.usa/a-comparison-between-mysql-vs-ms-sql-server-58b537e474be

Office of the Privacy Commissioner of Canada. (2020) The *Personal Information Protection and Electronic Documents Act* (PIPEDA).Retrieved from https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/

Palchuk, M. Uptech.(2020) Mapbox Vs Google Maps: What Maps API is Best For Your App?. Retrieved from https://uptech.team/blog/mapbox-vs-google-maps-vs-openstreetmap

Presentation Logic. Retrieved from https://www.sciencedirect.com/topics/computer-science/presentation-logic

Rosencrance, L. (2019). Explore multi-factor authentication products in-depth. Retrieved 9 June 2020, from https://searchsecurity.techtarget.com/feature/Explore-multifactor-authentication-products-in-depth?amp=1

Stackoverflow.(2010 December). What is message-oriented middleware. Retrieved from https://stackoverflow.com/questions/3975474/what-is-message-oriented-middleware
Streaming Data from MySQL into Kafka with Kafka Connect and Debezium. (2018, March 24). Retrieved from https://rmoff.net/2018/03/24/streaming-data-from-mysql-into-kafka-with-kafka-connect-and-debezium/

Strom, D. (2020). CA Strong Authentication offers businesses low-cost MFA and 2FA. Retrieved 9 June 2020, from https://searchsecurity.techtarget.com/feature/Multifactor-authentication-products-CA-Strong-Authentication

Square.(2020). Learn about Square's Fees. Retrieved from

https://squareup.com/help/ca/en/article/5068-what-are-square-s-fees#:~:text=Square's%20Payment%20Processing%20Fees,That's%20it.

MySQL. Retrieved 14 June 2020, from https://www.saasworthy.com/product/mysql

The Popularity of Google Maps: Trends in Navigation Apps in 2018 | The Manifest. Retrieved 14 June 2020,from https://themanifest.com/mobile-apps/popularity-google-maps-trends-navigation-apps-2018

Top U.S. mapping apps by users 2018 | Statista. Retrieved 14 June 2020, from https://www.statista.com/statistics/865413/most-popular-us-mapping-apps-ranked-by audience/#:~:text=Most%20popular%20mapping%20apps%20in%20the%20U.S.%202018%2C%20by%2audince&text=During%20this%20month%2C%20154.4%20million,include%20Waze%20and%20Google 20Earth

University of Toronto. (2020). *Quick facts*. Retrieved from https://www.utoronto.ca/about-u-of-t/quick-facts

What is Database Hosting? About Database Hosting. (2019, May 24). Retrieved from https://www.atlantic.net/what-is-database-hosting/

Wikipedia. (2020 March 7). Thin Client. Retrieved from https://en.wikipedia.org/wiki/Thin_Client

APPENDIX

Appendix I: User Account Management (Login and Validation)

Alternative	Cost Effective	Security	Integration compatibility	Scalability	Flexibility
CA Strong Authentication	***	***	***	***	***
Firebase Authentication	**	**	**	***	**
AuthPoint Authentication	***	***	**	**	**
Interout MFA	**	***	***	**	
Okta Adaptive Authentication	**	***	***	**	***

Appendix II: Instant Messaging SaaS

Alternatives	Cost Efficiency	Ban/Delete Users	Support	Message retrieval & data export
SendBird	*	***	***	*
	Starter: $4199 Pro: $6699 Enterprise: higher than above	Only available for Pro and Enterprise; also include profanity filters	Very limited support for Starter and Pro	N.A. for Starter and Pro, extra charge for Enterprise
Stream	***	*****	**	*

	Enterprise: > $499 (Standard Chat: $499 but only supports 25k MAU)	Spam & profanity protection, ban Users, flag message	Didn't specify for Standard, and the phone & slack support time for Enterprise	N.A.
Applozic	****	*****	****	***
	Pro Plan: $499	Also include profanity filters	SLA-Based (maybe shorter than 48 hrs) & dedicated account manager	Lifetime message history retention but didn't specify export options
Talk JS	*****	*	***	***
	Standard: $279, unlimited everything Premium: $569	Only has "mask contact information", didn't specify User mgmt. options	Limited support for Standard and Premium	Only has "dashboard with full conversation history", didn't specify export options

Appendix III: Data Store and Access

Alternatives	Cost Efficiency	Real-Time Data Manipulation	Speed & Performance	Scalability & Extensibility	Data Maintenance
MySQL	****	****	****	****	***
Microsoft SQL Server	****	***	****	***	***
Oracle	**	****	****	****	****
NO SQL (MongoDB)	*	*****	**	*	*

Appendix IV: Maps and location

Alternatives	Cost Efficiency	Accuracy	Location Tracking	Customization	Other Comments
Google Maps API	***	*****	*****	****	Heavy to render, drains battery
OpenStreetMap API	*****	***	**	****	Open-source database
Mapbox API	***	***	***	*****	Geared towards larger project

Appendix V: Payment Technology

Alternatives	Convenience	Cost Efficiency	Security	Ease of Integration	Speed
Integrated Payment with Paypal	Users require account	3.9% +0.30 per transaction	Secure	Customer is redirected to Paypal portal, App Integration is extra 30 dollars per month and a custom quote for transaction fees.	3-5 business days
Integrated Payment with Square	Users don't require account	3.4%+0.15	Secure with fraud detection+end to end encryption	Customizable UI	1-2 business days
Integrated with BrainTree	Users don't require account	2.9%+0.30	Secure with end to end encryption	Customizable UI	2 business days