

Final Project Report

Louie Venegas

CEIS200

October 21, 2024

1 Contents

1	IEEE 830 Software Requirements Specifications.....	5
1.1	Introduction.....	5
1.1.1	Purpose.....	5
1.1.2	Document Conventions.....	5
1.1.3	Intended Audience and Reading Suggestions.....	6
1.1.4	Product Scope.....	6
1.1.5	References.....	7
1.2	Overall Description.....	7
1.2.1	Product Perspective.....	7
1.2.2	Product Functions.....	7
1.2.3	User Classes and Characteristics.....	7
1.2.4	Operating Environment.....	8
1)	Design - Web browsers (for Blazor server-side application).....	8
2)	iOS and Android devices (for MAUI mobile application.....	8
3)	Azure cloud environment for backend services and Implementation Constraints.....	8
1.2.5	User Documentation.....	8
1.2.6	Assumptions and Dependencies.....	8
1.3	External Interface Requirements.....	9
1.3.1	User Interfaces.....	9
1.3.2	Hardware Interfaces.....	9
1.3.3	Software Interfaces.....	9
1.3.4	Communications Interfaces.....	10
1.4	System Features.....	10
1.4.1	System Feature #1.....	10
1.5	System Feature #2.....	10
1.5.1	System Feature #3.....	10
1.5.2	System Feature #4.....	11
1.5.3	System Feature #5.....	11
1.5.4	System Feature #n (add system features as necessary).....	11
1.6	Other Nonfunctional Requirements.....	11
1.6.1	Performance Requirements.....	11
1.6.2	Security Requirements.....	11
1.6.3	Software Quality Attributes.....	12
1.6.4	Business Rules.....	12
2	OOAD Diagrams.....	13
2.1	Updated Use Case Diagram.....	13
2.2	Use Case Specifications.....	14

1.7	1.2.1 Use Case #1:	14
1.8		14
1.9	1.2.2 Use Case #2:	15
1.10		16
1.11		16
1.12		16
1.13		16
1.14	1.2.3 Use Case #3:	16
1.15		17
1.16		17
1.17		17
1.18	1.2.4 Use Case #4:	17
1.19	1.2.5 Use Case #5:	18
1.20	1.2.6 Use Case #6:	19
1.21	1.2.7 Use Case #7:	20
1.22	1.2.8 Use Case #8:	21
1.23	1.2.9 Use Case #9:	23
1.24	1.2.10 Use Case #10:	24
1.25		25
1.26	1.2.11 Use Case #11:	25
1.27	1.2.12 Use Case #12:	26
1.28	1.2.13 Use Case #13:	27
1.29		28
2.3	Class Diagram	29
2.4	Sequence Diagrams	29
2.5	State Machine Diagram	32
3	Risk Assessment Matrix	33
4	Updated Project Plan	34
5	Test Plan	34
5.1	Test Plan Identifier	34
5.2	Introduction(scope):	34
5.3	Test Items	34
10.1	Features to be Tested	35
10.2	Features not to be Tested	35
10.3	Approach	35
10.4	Items Pass/Fail Criteria	35
10.5	Test Deliverables	36
10.6	Environmental Needs	36
10.7	Responsibilities	37

10.8	Staffing and Training Needs.....	37
10.9	Schedule.....	38
10.10	Risks and contingencies.....	38
2		39
11	Test Cases.....	39
11.1	Test Cases.....	39
12	Updated Project Plan.....	42

1 IEEE 830 Software Requirements Specifications

1.1 Introduction

1.1.1 Purpose

This project's purpose is to develop a mobile and web-based platform that addresses challenges faced by food truck/stand owners and customer/enthusiast and establishes a sustainable business model. The purpose of the platform can be outlined in the following points.

- 1) **Create a community** for food truck/stand culture, to foster connections between vendors and customer/enthusiast.
- 2) **Develop consistency and accountability** of food trucks/stands through a process of establishing menus and pricing.
- 3) **Improve customer/enthusiast experience** by providing real-time location tracking and event information for food trucks/stands.
- 4) **Implement a loyalty program** to reward frequent customers/enthusiast and have an incentive program in place to reward vendor participation.
- 5) **Facilitate events** by planning and encouraging coordination between food truck/stand owners encouraging organized gatherings in order promote growth within the industry and drive maximize profits for the food truck/stands.
- 6) **Offer a tiered system for vendors**, providing various levels of engagement and features to suit different business needs.
- 7) **Establish a user-driven rating and review system** to maintain quality standards and inform customers/enthusiast about their dining choices.
- 8) **Grow a scalable business model** through multiple revenue streams, including:
 - a) Tiered subscription plans for vendors
 - b) Advertising opportunities
 - c) In-app purchases
 - d) Event ticketing and coordination services
 - e) Payment gateway integration for transactions

These points will guide the platform in streamlining operations for vendors, enhance the dining experience for customer/enthusiast, contribute to the sustainable growth of the food truck/stand culture, and generate revenue to ensure the long-term viability and continued development of the platform.

1.1.2 Document Conventions

Terminology:

- 1) "Customer/Enthusiast" refers to end-users of the platform who are not food truck/stand owners.
- 2) "Food Truck/Stand" encompasses all types of mobile food vendors.

1.1.3 Intended Audience and Reading Suggestions

Project Managers, Developers, Quality Assurance QA Testers, Business Analyst, Stakeholders(clients, end users anyone with a personal stake in the project), Technical Writers, UX/UI Designers, System Architects, Operations Team, security analyst and Compliance. Officers

1.1.4 Product Scope

This Software Requirements Specification will define the requirements for a mobile and web-based platform designed to connect food truck/stand owners with customers/enthusiasts.

The scope of this document includes:

- 1) **User Management System:** Specifications for user registration, authentication, and profile management for both vendors and customers. This includes the implementation of a tiered system for vendors with varying levels of features and engagement.
- 2) **Community Platform:** Requirements for social networking features, messaging systems, and forums to foster connections within the food truck/stand community.
- 3) **Food Truck/Stand Management:** Specifications for menu and pricing management, real-time location tracking, and operating hours management to improve consistency and accountability.
- 4) **Event Planning and Coordination:** Requirements for event creation, scheduling, discovery, and coordination tools to facilitate organized gatherings and industry growth.
- 5) **Loyalty and Reward Program:** Specifications for customer loyalty programs and vendor incentive systems, including points tracking and redemption mechanisms.
- 6) **Review and Rating System:** Requirements for a user-driven review and rating system to maintain quality standards and inform customer choices.
- 7) **Search and Discovery Features:** Specifications for search capabilities for food trucks/stands, menu items, and events.
- 8) **Notification System:** Requirements for push notifications, email alerts, and in-app messaging to keep users informed of relevant updates.
- 9) **Payment and Transaction System:** Specifications for payment gateway integration, in-app purchases, and subscription management for vendors.
- 10) **Advertising Platform:** Requirements for ad creation, management, targeting, and analytics tools to support the business model.
- 11) **Performance and Security:** Specifications for system performance, data security, and user privacy protection.
- 12) **Interfaces:** Requirements for user interfaces (mobile and web), as well as any necessary hardware, software, and communication interfaces.

This Requirements Specification will provide an extensive guide for the development team and assist in the creation of a platform that enhances the food truck/stand industry(advertisement and

maximizing profitability through collaboration), improves customer/enthusiast experiences, and establishes a sustainable business model.

1.1.5 References

"IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286. keywords: {Software requirements and specifications;contract;customer;prototyping;software requirements specification;supplier;system requirements specifications},

Pressman, R. S. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education

1.2 Overall Description

1.2.1 Product Perspective

The Food Truck/ Community Platform is a solution designed with intent to connect food truck/stand owners with customers/enthusiasts. This platform will operate as a web application and a mobile app, integrated with cloud-based services to provide real-time features and scalable performance.

1.2.2 Product Functions

- 1) User management with tiered access
- 2) Real-time location tracking and mapping(for android integrate google maps for apple integrate apple maps via API interfaces)
- 3) Event planning and coordination
- 4) Loyalty, rewards and incentive programs(each will need business rules to govern there functions)
- 5) Customer/enthusiast review and rate the food truck/stand(options to follow and like) negative comments can be buried with good ones over time the rating will always be an average.
- 6) Search and discovery features
- 7) Payment processing and integration(part of the business model)
- 8) Advertising and promotional tools
- 9) Social media integration
- 10) Multi language support to maximize our customer base
- 11) In-app help system customer support ticketing and a place file formal complaints against vendors
- 12) Notifications(push and email)

1.2.3 User Classes and Characteristics

- 1) **Vendors(food truck/ stand owners):** tiered memberships
- 2) **Customers/enthusiasts:** always free accounts

1.2.4 Operating Environment

- 1) **Design** - Web browsers (for Blazor server-side application)
- 2) **iOS and Android** devices (for MAUI mobile application)
- 3) **Azure cloud** environment for backend services and Implementation Constraints

1.2.5 User Documentation

The platform will provide documentation user documentation in the following areas but not limited to only these areas:

- 1) **User Manuals:** Separate guides for vendors and customers
- 2) **In-App Help:** Context-sensitive help and tooltips
- 3) **Knowledge Base:** Searchable database of articles and FAQs
- 4) **Video Tutorials:** Short, task-specific guides
- 5) **API Documentation:** For developers integrating with the platform

All documentation will be version-controlled and available in multiple languages to support a diverse and inclusive user base.

1.2.6 Assumptions and Dependencies

Assumptions:

- 1) Reliable internet connectivity for real-time features
- 2) Vendor adoption of the platform
- 3) Compliance with mobile app store guidelines
- 4) Successful partnership with a major social media platform for targeted advertising and influencer marketing campaigns to drive user acquisition and engagement

Dependencies:

- 1) Microsoft .NET 6.0 or later for Blazor and MAUI development
- 2) Azure SDK for cloud service integration
- 3) Entity Framework Core for database operations
- 4) SignalR for real-time communications
- 5) Third-party APIs: Google Maps or Mapbox for location services
- 6) Payment gateway APIs (e.g., Stripe, PayPal)
- 7) Azure Cosmos DB SDK for NoSQL database operations
- 8) OAuth 2.0 provider (e.g., Azure Active Directory B2C) for secure user authentication and authorization
- 9) OpenID Connect for seamless single sign-on capabilities.
- 10) Social media platform API for integrated marketing campaigns

The project's success relies on the availability and compatibility of these technologies and services throughout the software development lifecycle, as well as effective execution of the marketing strategy to drive platform adoption.

1.3 External Interface Requirements

1.3.1 User Interfaces

The user interface will prioritize ease of use, visual appeal and efficient access to key features. The goal in the design of the user interface is to enhance the experience for both the food truck/stand vendors and customer users.

Some general principles will be responsive design for compatibility across devices and screen size, easy to use or familiar navigation and consistency across pages.

- 1) **Header:** Logo, Main navigation menu, User account access/login, Search functionality
- 2) **Dashboard:** Metrics for vendors, Personalized feed for customers, Quick access to frequently used features
- 3) **Footer:** Quick links to key sections, social media links, Legal information and privacy policy
- 4) **mobile application:** will include all features but also include maps, and push notifications, pull to refresh, gesture controls.
- 5) **Map View:** interactive map showing food truck locations, Filters for cuisine type, price range, and ratings.
- 6) **Food Truck/Stand Vendor Profiles:** Photo gallery, Menu display, Reviews and ratings, Location and hours information
- 7) **Event Creation/Management (for vendors):** Calendar interface, Event details form, Attendee management tools
- 8) **Notifications and Alerts:** in-app notification center, Push notification support for mobile devices, Email notifications for key actions and updates

1.3.2 Hardware Interfaces

The system will be designed to function primarily through standard mobile devices, with minimal reliance on specialized hardware. Core functionality will be maintained across different devices and network conditions.

1.3.3 Software Interfaces

The Food Truck/Stand Community Platform will interface with the following key software systems:

- 1) **Database Management System** to store and manage all user data, food truck information, and events.
- 2) **Authentication System** Azure Active Directory B2C To manage user authentication across the platform.

- 3) **Payment Gateway:** Payment Processing API with a direct integration with the platform, requiring no additional user accounts it should Support for major credit/debit card payments. The purpose of this gateway is to facilitate payments between the vendors and the platform. At this time, the platform will not facilitate orders from customers to vendors.
- 4) **Mapping and Location Services** Google Maps API (Android) and Apple Maps API (iOS) To provide real-time location tracking and mapping.
- 5) **Push Notification Services interface** to send real-time notifications to mobile app users.

All software interfaces will use secure protocols while adhering to industry-standard security practices.

1.3.4 Communications Interfaces

The Food Truck/Stand Community Platform will utilize various communication protocols for smooth operation and data exchange. Such as but not limited to:

- 1) **HTTPS Protocol:** To facilitate secure communication between the clients and the server, HTTPS will ensure encryption and data integrity so all data transmission should use this protocol.
- 2) **WebSocket Protocol:** Enabling bidirectional communication for features like live chat and instant notifications, push notifications and real-time updates on the web application.
- 3) **REST API:** for communication between the front-end applications and the back-end services or client and server following REST architectural principles.
- 4) **Mobile Push Notification Services:** To send notifications to mobile devices even when the app is not actively running.
- 5) **SMS Gateway:** for sending critical notifications or two-factor authentication codes via text message.

All communication interfaces will adhere to industry-standard security practices.

1.4 System Features

1.4.1 System Feature #1

User Management System

- 1) user registration
- 2) profiles
- 3) authentication.

1.5 System Feature #2

Real Time Location Tracking

- 1) Integration with system mapping services
- 2) Vendors locations

- 3) Notify customers of nearby vendors

1.5.1 System Feature #3

Event Planning and Coordination

- 1) Allow Vendors to create events
- 2) Event discovery system
- 3) RSVP features
- 4) Event notifications

1.5.2 System Feature #4

Vendor Review and Rating System

- 1) Allows customers to rate vendors on a 5-star scale
- 2) Display average review for each vendor
- 3) Implement a feature for customers to post reviews
- 4) Moderation system to remove inappropriate reviews
- 5) Vendor response features for reviews

1.5.3 System Feature #5

Menu Management

- 1) All vendors must submit a menu this is mandatory and always editable
- 2) Allow vendors to post pictures
- 3) Vendors can turn on and off availability for items
- 4) Restrict price updates to once a week

1.5.4 System Feature #n (add system features as necessary)

1.6 Other Nonfunctional Requirements

1.6.1 Performance Requirements

The following requirements are preliminary and subject to adjustment based on performance testing, infrastructure capabilities, and specific project needs. Final requirements will be determined through iterative refinement and stakeholder consultation.

- 1) The system shall load all pages within 3 seconds under normal network conditions.
- 2) Database queries shall return results within 1 second for 95% of requests.
- 3) The mobile app shall consume no more than the minimum amount of device storage than necessary for operations.

1.6.2 Security Requirements

- 1) The system shall enforce multi-factor authentication for all user accounts.
- 2) All data transmissions shall be encrypted using the latest encryption standards

- 3) The system shall automatically log out inactive users after 15 minutes of inactivity except on trusted devices or when the preference has been changed.
- 4) The system should defend itself against brute-force attacks.
- 5) The system should use sanitization and validation to prevent SQL injection attacks.
- 6) Password policies are to be strong

Security Requirements in the Software Development Lifecycle (SDLC)

- 7) **Planning Phase:** Establish a security budget and allocate resources for security measures.
- 8) **Requirements Phase:** Define specific security requirements in the project documentation.
- 9) **Design Phase :** Plan for secure data storage and transmission using encryption.
- 10) **Development Phase:** Implement secure configurations for all components in the production environment.
- 11) **Maintenance phase:** Conduct ongoing security assessments

1.6.3 Software Quality Attributes

- 1) Reliability
- 2) Usability
- 3) Efficiency
- 4) Maintainability
- 5) Portability
- 6) Scalability
- 7) Availability

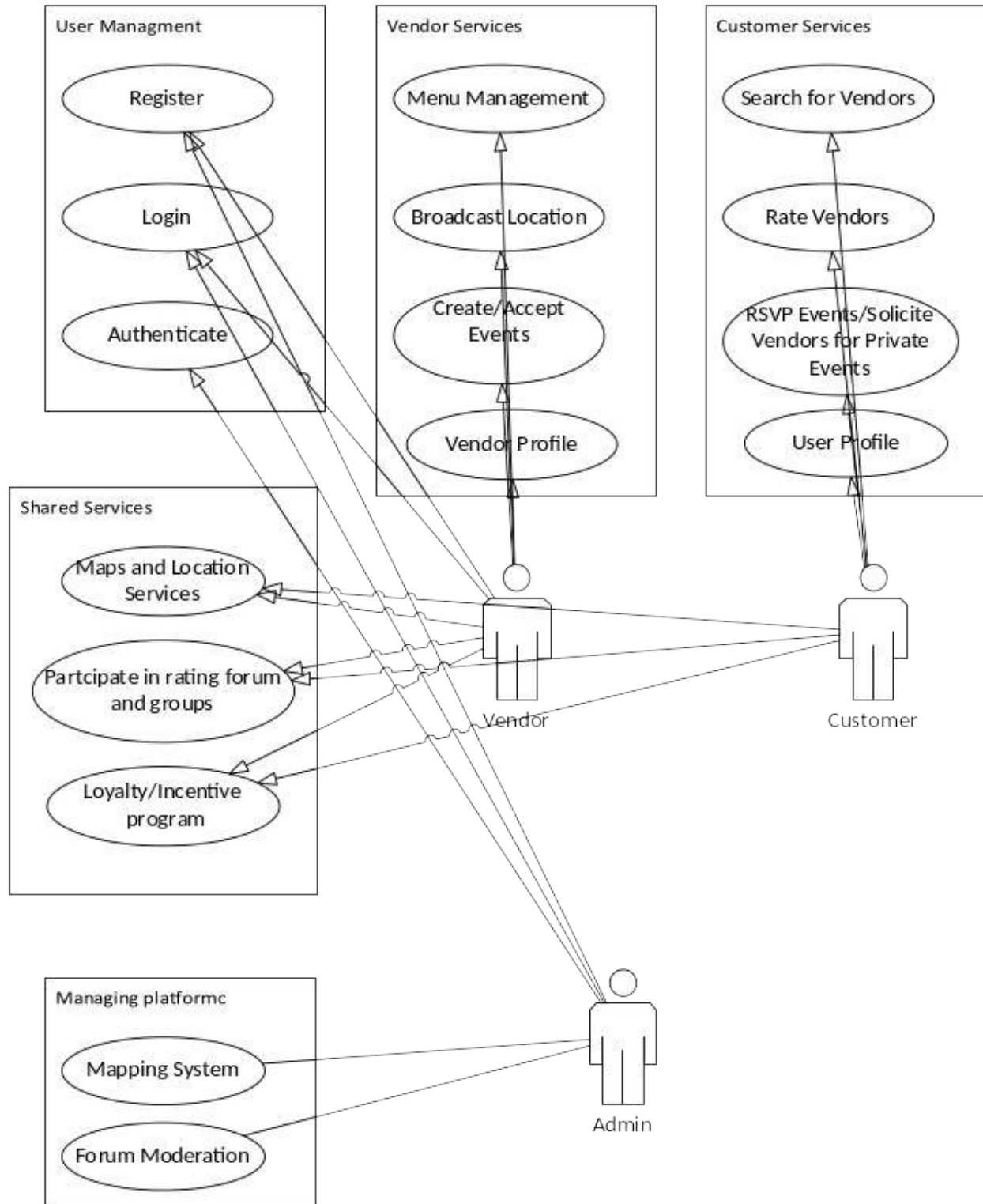
1.6.4 Business Rules

The following categories of business rules have been identified. Specific rules within these categories will be developed by the development team.

- 1) User registration and access rules
- 2) Vendor tiered access rules
- 3) Location tracking and display rules
- 4) Event creation and management rules
- 5) Review and rating guidelines
- 6) Menu management policies
- 7) Loyalty program rules
- 8) Data retention and privacy policies
- 9) Content moderation guidelines and

2 OOAD Diagrams

2.1 Updated Use Case Diagram



2.2 Use Case Specifications

1.7 1.2.1 Use Case #1:

1.8

Use Case Name:	Registration	ID: 1	Importance Level: 1
Primary Actor:	customer/vendor	Use Case Type:	System
Stakeholders and Interests: customers/vendors/bank			
Brief Description: when a vendor or customer signs up to the platform they must register.			
Trigger:	user initiation		
Relationships: Association: Login, Authentication, credential verification, profile			
Normal Flow of Events: 1. User initiates the process of registration 2. User provides mandatory information and credentials(Vendors will have a stricter standard then the customers for obvious business reasons), for customers mandatory information is sufficient for vendors identification documentation will be needed. 3. Information provided is verified and account profile is created			
SubFlows: S-1: Verification of customer 1. Provides personal information and contact information. 2. Contact information is verified in some way 3. Two step authentication is established. S-2: Verification of vendor 1. Provides personal information, contact information personal identification and business information(Obviously this wont be a strict enforcement for business verification dealing with Food truck/stands but at least a business name and type of food served). 2. Contact information is verified in some way personal identification will be verified in some way. 3. Two step authentication established. S-3: Profile Creation 1. After registration and two step verification is enforced profile template are assigned and allowed to be customized.			
Alternate/Exceptional Flows: none noted			

1.9 1.2.2 Use Case #2:

Use Case Name: Login	ID: 2	Importance Level: 1
Primary Actor: customer/vendor/admin	Use Case Type: System	
Stakeholders and Interests: all users and staff		
Brief Description: Upon successful completion of registration the user will be able to log into the web platform and mobile application		
Trigger: User initiates a login event		
Relationships: Association: Profile, Users, authentication		
Normal Flow of Events: 1. User initiates a login event 2. User puts in credentials and clears two step authentication 3. Users is presented with home page		
SubFlows: S-1: credentials verification 1. User provides credentials 2. If found user then is passed to two step verification S-2: two step authentication 4. User is presented with the options available for two step verification and chooses one. 5. User provides the required information and is logged into the user account		
Alternate/Exceptional Flows: If the user can not provide valid credentials after X amount of attempts they are offered a way to recover the lost credentials. If this can not be accomplished then no access is to be granted.		

1.10

1.11

1.12

1.13

1.14 1.2.3 Use Case #3:

Use Case Name:	Authentication	ID: 3	Importance Level: 1
Primary Actor:	all users	Use Case Type:	System
Stakeholders and Interests: all users			
Brief Description:			
Authentication is required during a qualifying action. A qualifying action being a login, a transaction, registering a public event and soliciting vendors for private events			
Trigger: Login attempt or qualifying action			
Type: Security			
Relationships:			
Association: logins, transactions, events			
Normal Flow of Events:			
User attempts a qualifying action			
System triggers two-step verification and requests verification			
User provides verification			
System authorizes the action			
SubFlows:			
1. none			
Alternate/Exceptional Flows:			
A-1: Verification Failure			
1. User fails to provide correct verification			
2. System rejects the action			

1.15

1.16

1.17

1.18 1.2.4 Use Case #4:

Use Case Name:	Menu Management	ID: 4	Importance Level: 2
Primary Actor:	Vendors	Use Case Type:	Abstract
Stakeholders and Interests: Customers and Competition			
Brief Description: In order for a vendor to be able to use the public services available on the platform they have to establish a brief menu with prices.			
Trigger: Vendor decides to set up or update their menu on the platform Type: User-initiated			
Relationships: Association: Vendor Include: User Authentication, Item Price Setting Generalization: None			
Normal Flow of Events: Vendor logs into their account on the platform Vendor navigates to the "Menu Management" section Vendor selects "Create New Menu" or "Edit Existing Menu" Vendor adds menu items, including name, description, and price for each item Vendor organizes items into categories if applicable Vendor reviews the complete menu Vendor submits the menu for publication on the platform System validates the menu data System confirms successful menu publication			
SubFlows: S-1: Vendor selects "Add New Item" Vendor enters item details (name, description, price, category) Vendor saves the new item S-2: Vendor selects an existing menu item Vendor modifies the item details as needed Vendor saves the changes			
Alternate/Exceptional Flows: A-1: Menu validation fails			

System displays error messages for invalid entries
 Vendor corrects the errors
 Vendor resubmits the menu for publication

A-2: Vendor cancels menu creation/edit

Vendor selects "Cancel" at any point during the process
 System prompts for confirmation
 If confirmed, system discards unsaved changes and returns to the main menu management screen

1.19 1.2.5 Use Case #5:

Use Case Name:	Broadcast Location	ID: 5	Importance Level: 1
Primary Actor:	Vendor	Use Case Type:	system
Stakeholders and Interests: Vendors, customers, platform			
Brief Description: Vendor broadcast location on map for customers to see where they are at			
Trigger: Vendor initiates location broadcast or system automatically updates vendor's location			
Type: User-initiated and System-initiated			
Relationships: Association: Vendor Include: GPS Location Services, Map Integration Generalization: None			
Normal Flow of Events: Vendor logs into the mobile application System requests permission to access device's location Vendor grants permission to access location System retrieves current GPS coordinates System updates vendor's location on the platform's map System confirms to vendor that location has been broadcast Customers can now see vendor's location on their map view			
SubFlows: S-1: Vendor selects "Update Location" option System retrieves current GPS coordinates System updates vendor's location on the map S-2: Vendor selects "Set Broadcast Duration" Vendor chooses a time period (e.g., 1 hour, 4 hours, until manually stopped) System sets a timer for the chosen duration			
Alternate/Exceptional Flows:			

Vendor wants to stop broadcasting
Vendor selects "Stop Broadcasting" option
System confirms vendor's choice
System removes vendor's location from the public map
Automatic location update
System detects significant change in vendor's location
System automatically updates vendor's location on the map
System notifies vendor that location has been updated

1.20 1.2.6 Use Case #6:

Use Case Name: Profiles	ID: 6	Importance Level: 1
Primary Actor:	Use Case Type: system	
Stakeholders and Interests: vendors, customers, platform		
Brief Description: After account creation user is prompted to either make a customer profile or a vendor profile and allowed to customize pictures and various things. Customers get a home page and a news feed Vendors get a landing page for advertising menus and brand.		
Trigger: Account creation completion Type: System-initiated		
Relationships: Association: User (Customer or Vendor) Include: Account Creation, Image Upload Extend: Menu Management (for Vendors) Generalization: None		
Normal Flow of Events: User completes account creation System prompts user to choose profile type User selects Customer or Vendor System presents appropriate profile form User fills information and uploads images System validates entered data System saves profile and generates landing page User is directed to new profile/landing page		
SubFlows: S-1: Customer Profile 1. User enters name, location, preferences 2. User sets news feed preferences 3. System generates personalized home page		

S-2: Vendor Profile 1. User enters business name, type, description 2. User uploads logo and brand images 3. User sets up initial menu or is prompted to 4. System generates vendor landing page
Alternate/Exceptional Flows: A-1: Invalid information 1. System highlights errors 2. User corrects and resubmits A-2: Skip customization 1. System creates basic profile 2. Prompts completion at next login A-3: Change profile type 1. System warns of data loss 2. If confirmed, starts new profile process 3. Transfers applicable data

1.21 1.2.7 Use Case #7:

Use Case Name:	Vendor search	ID: 7	Importance Level:
Primary Actor:	Use Case Type:		
Stakeholders and Interests: Customers: Find relevant local vendors quickly Vendors: Be discoverable by potential customers Platform: Provide efficient search functionality			
Brief Description: customers can search for local vendors based on filters such as cuisine type and distance from real time location and other customer ratings			
Trigger: Customer initiates vendor search Type: User-initiated			
Relationships: Association: Customer Include: Location Services, Rating System Extend: Vendor Profile View Generalization: None			
Normal Flow of Events: Customer selects "Search Vendors" option System requests customer's current location Customer provides search criteria (cuisine, distance, etc.) System retrieves and displays matching vendors Customer views search results			

SubFlows: S-1: Apply Filters Customer selects additional filters System updates search results in real-time S-2: Sort Results Customer chooses sorting criteria (rating, distance, etc.) System reorders search results accordingly
Alternate/Exceptional Flows: A-1: No results found System displays "No matching vendors" message Suggests broadening search criteria A-2: Location services unavailable System prompts customer to enter a location manually Proceeds with search based on entered location

1.22 1.2.8 Use Case #8:

Use Case Name:	rate vendors	ID: 8	Importance Level: 2
Primary Actor:	customer	Use Case Type:	system
Stakeholders and Interests: Customers: Share experiences and feedback Vendors: Receive feedback and maintain reputation Platform: Ensure quality service and user engagement			
Brief Description: Customers can rate vendors and leave comments on a discussion board vendors can respond to comments made.			
Trigger: Customer decides to rate a vendor Type: User-initiated			
Relationships: Association: Customer, Vendor Include: Vendor Profile View Extend: Vendor Search Generalization: None			
Normal Flow of Events: 1. Customer selects "Rate Vendor" option on vendor's profile 2. System displays rating form and comment section 3. Customer submits rating and comment 4. System validates and saves the review			

5. System updates vendor's overall rating 6. System notifies vendor of new review
SubFlows: S-1: Vendor Response Vendor views new review notification Vendor submits response to customer's comment System adds response to the discussion board S-2: Edit/Delete Review Customer selects option to edit/delete their review System allows modification or removal of the review System updates vendor's overall rating if necessary
Alternate/Exceptional Flows: A-1: Invalid Rating Submission System detects invalid input (e.g., no rating selected) System prompts customer to correct the input Customer corrects and resubmits A-2: Inappropriate Content System flags potentially inappropriate content Moderator reviews flagged content Moderator decides to approve, edit, or remove content

1.23 1.2.9 Use Case #9:

Use Case Name:	maps and location services	ID: 9	Importance Level: 1
Primary Actor:	system	Use Case Type:	System
Stakeholders and Interests: Vendors: Broadcast real-time location Customers: Find vendors easily Platform: Provide accurate location-based services			
Brief Description: Maps and location services use real-time location tracking for broadcasting vendor location and helping customers find vendors.			
Trigger: System initialization or user request Type: System-initiated and User-initiated			
Relationships: Association: Vendor, Customer Include: GPS Services, Map Integration			

Extend: Vendor Search, Vendor Profile Generalization: None
Normal Flow of Events: 1. System initializes location services 2. System requests location permissions from users 3. System tracks vendor locations in real-time 4. System updates vendor positions on map 5. Customer requests nearby vendor locations 6. System displays vendors on customer's map
SubFlows: S-1: Vendor Location Broadcast Vendor activates location broadcasting System continuously updates vendor's location S-2: Customer Navigation Customer selects a vendor System calculates route from customer to vendor System provides turn-by-turn directions
Alternate/Exceptional Flows: A-1: Location Services Unavailable System detects location services are off System prompts user to enable location services If user declines, system uses last known location A-2: Vendor Goes Offline System detects vendor's device is offline System marks vendor as "Currently Unavailable" System notifies customers searching for this vendor

1.24 1.2.10 Use Case #10:

Use Case Name:	forums/groups/discussions	ID: 10	Importance Level: 1
Primary Actor:	User (Customer or Vendor)	Use Case Type:	Primary
Stakeholders and Interests: Users: Participate in community discussions, share information Platform Administrators: Ensure proper use, moderate content Business: Increase user engagement and retention			
Brief Description: Users should be able to create and participate in forums groups and community discussions			
Trigger: User accesses community features			

Type: User-initiated
Relationships: Association: User Profile Include: Content Creation, Notification System Extend: User Search, Reporting System Generalization: None
Normal Flow of Events: 1. User navigates to community section 2. System displays available forums/groups/discussions 3. User selects a forum/group/discussion or creates a new one 4. System loads the selected content or creation form 5. User participates by posting, commenting, or moderating
SubFlows: S-1: Create New Forum/Group/Discussion User selects "Create New" option User fills out required information (title, description, etc.) System validates and creates the new entity S-2: Moderate Content Moderator reviews flagged content Moderator decides on appropriate action (approve, edit, remove) System updates content based on moderator's decision
Alternate/Exceptional Flows: A-1: Inappropriate Content Detected System or user flags potentially inappropriate content System notifies moderators Content is hidden pending review A-2: User Leaves Group User selects option to leave group System confirms user's decision System removes user from group and updates permissions

1.25

1.26 1.2.11 Use Case #11:

Use Case Name:	loyalty program	ID: 11	Importance Level: 1
Primary Actor:	Customer, Vendor	Use Case Type:	fully dressed
Stakeholders and Interests:			

Customers: Earn rewards for frequent use of the platform Vendors: Increase customer retention and engagement Platform: Encourage user activity and loyalty
Brief Description: Implement a loyalty system for customer and incentive for vendors.
Trigger: Customer makes a purchase or vendor completes a transaction Type: System
Relationships: Association: Customer Profile, Vendor Profile Include: Transaction Processing, Reward Calculation Extend: Customer Support, Marketing Campaigns Generalization: None
Normal Flow of Events: 1. System detects a qualifying action (purchase, review, etc.) 2. System calculates loyalty points or vendor incentive 3. System updates user's loyalty status 4. System notifies user of new points/status 5. User can view updated loyalty information in profile
SubFlows: S-1: Redeem Rewards (Customer) Customer selects "Redeem Rewards" option System displays available rewards Customer chooses a reward System deducts points and issues reward S-2: Vendor Incentive Program System reviews vendor's performance metrics System calculates vendor's tier in incentive program 1. System applies relevant benefits (e.g., reduced fees, priority listing)
Alternate/Exceptional Flows: A-1: Insufficient Points for Redemption Customer attempts to redeem reward without sufficient points System displays error message System suggests ways to earn more points A-2: Vendor Performance Drop System detects a significant drop in vendor's performance System issues warning to vendor If not improved, system adjusts vendor's incentive tier

1.27 1.2.12 Use Case #12:

Use Case Name:	Event Creation	ID: 12	Importance Level: 1
Primary Actor:	Use Case Type: Business		
Stakeholders and Interests:			
Users: Create events and find suitable vendors			
Vendors: Receive invitations and opportunities to sell at events			
Platform: Facilitate event organization and vendor-customer connections			
Brief Description: users should be able to create events and invite or solicit vendors to attend and sell food.			
Trigger: User initiates event creation process			
Type: User-initiated			
Relationships:			
Association: User Profile, Vendor Profile			
Include: Location Services, Messaging System			
Extend: Vendor Search, Payment Processing			
Generalization: None			
Normal Flow of Events:			
1. User selects "Create Event" option 2. System displays event creation form 3. User fills in event details (date, location, type, etc.) 4. User specifies vendor requirements 5. System saves event information 6. System searches for matching vendors 7. User reviews and selects vendors to invite 8. System sends invitations to selected vendors			
SubFlows:			
S-1: Vendor Response			
Vendor receives event invitation			
Vendor reviews event details			
Vendor accepts or declines invitation			
System updates event organizer			
S-2: Event Modification			
User selects option to modify event			
System displays current event details			
User makes necessary changes			
1. System updates event information and notifies relevant parties			
Alternate/Exceptional Flows:			
A-1: No Suitable Vendors Found			
System fails to find vendors matching requirements			
System notifies user of the issue			
Suggests broadening search criteria or modifying event details			

A-2: Event Cancellation

User chooses to cancel the event

System prompts for confirmation

If confirmed, system cancels event and notifies all invited vendors

1.28 1.2.13 Use Case #13:

Use Case Name:	Vendor Solicitation/event creation	ID: 13	Importance Level: 1
Primary Actor:		Use Case Type:	business
Stakeholders and Interests:			
Vendors: Organize events, participate in events, get hired for private events			
Customers: Attend public events, hire vendors for private events			
Platform: Facilitate event organization and vendor-customer connections			
Brief Description:			
Vendors are allowed to organize public events and gatherings in order to bring multiple vendors together for organized events. Customers can solicit or hire vendors for private events.			
Trigger: Vendor initiates event creation or Customer initiates vendor solicitation			
Type: User-initiated			
Relationships:			
Association: Vendor, Customer			
Include: Vendor Profile, Location Services			
Extend: Messaging System			
Generalization: None			
Normal Flow of Events:			
1. User (Vendor or Customer) selects event creation/solicitation option 2. System presents event creation/solicitation form 3. User fills in event details or vendor requirements 4. System validates and saves the information 5. System publishes event or sends solicitation to relevant vendors			
SubFlows:			
S-1: Public Event Creation (Vendor)			
Vendor specifies event details (date, location, participating vendors)			
System notifies and confirms other vendors' participation			
System creates public event listing			
S-2: Private Event Solicitation (Customer)			
Customer specifies event requirements			
System matches requirements with available vendors			
1. System notifies matching vendors of the opportunity			
Alternate/Exceptional Flows:			

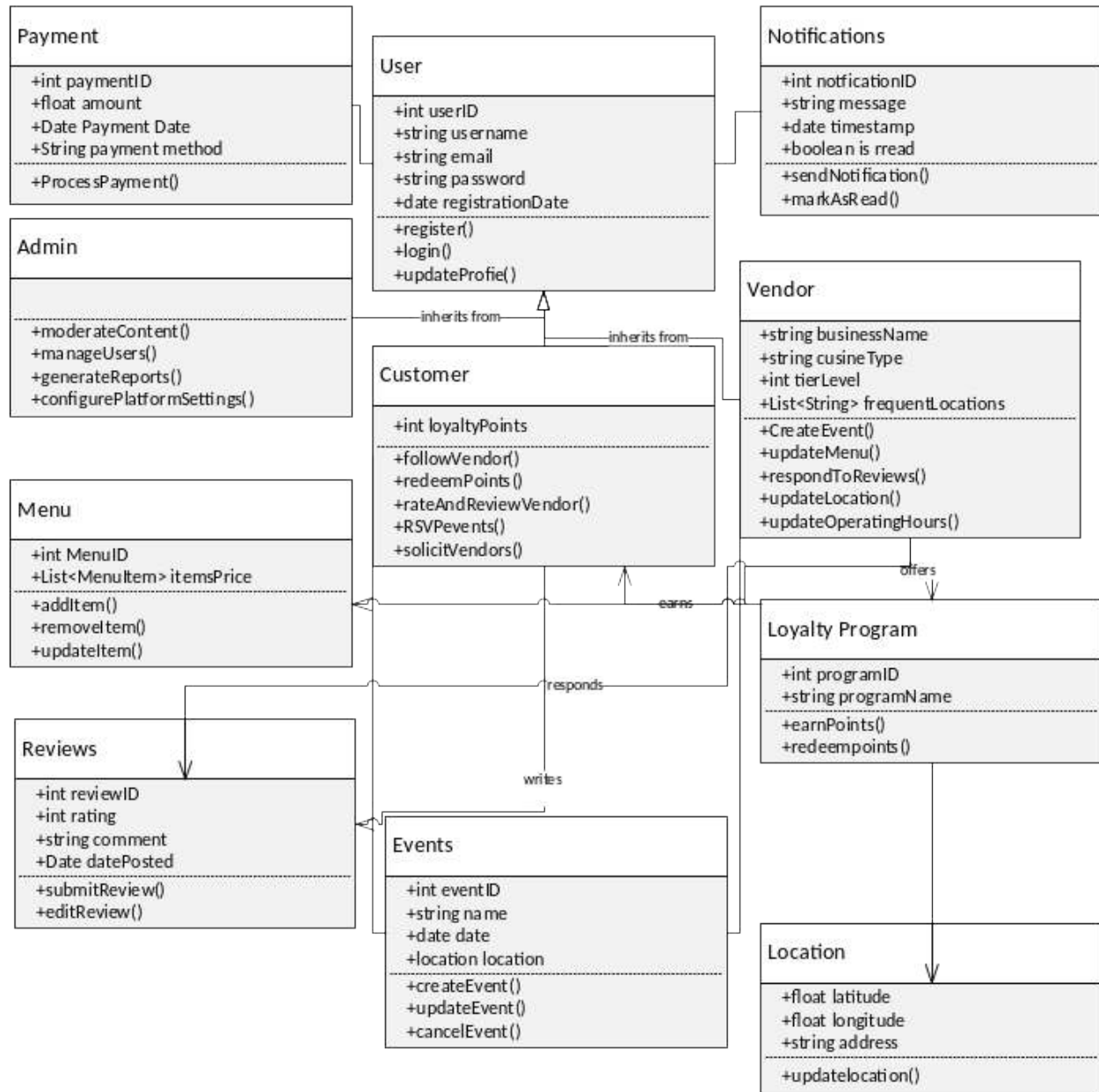
A-1: Event Cancellation

Creator cancels the event
System notifies all participants
System removes event listing

A-2: Vendor Declines Participation

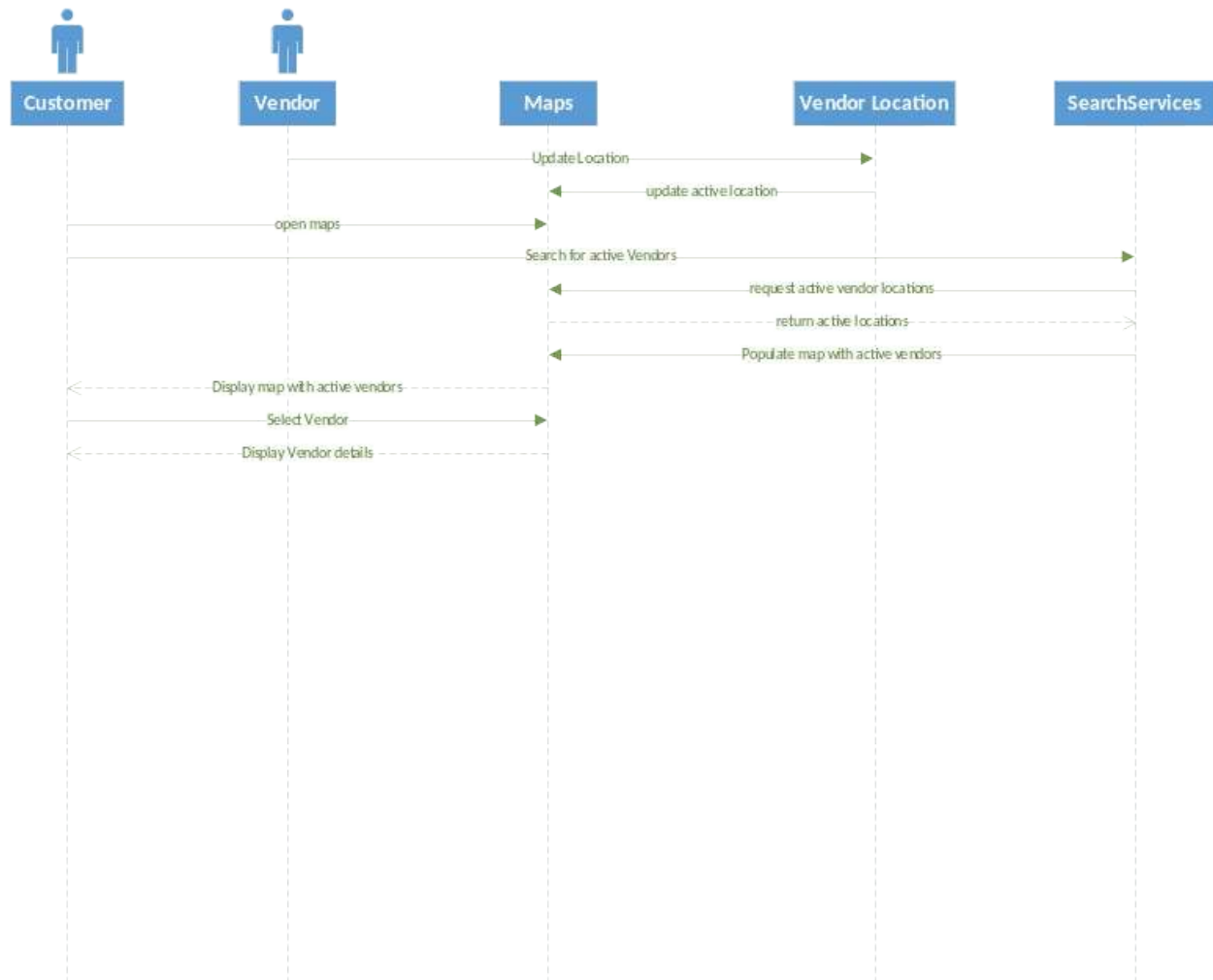
Vendor declines event participation or private event opportunity
System notifies event creator
Creator can choose to modify event or find replacement

2.3 Class Diagram

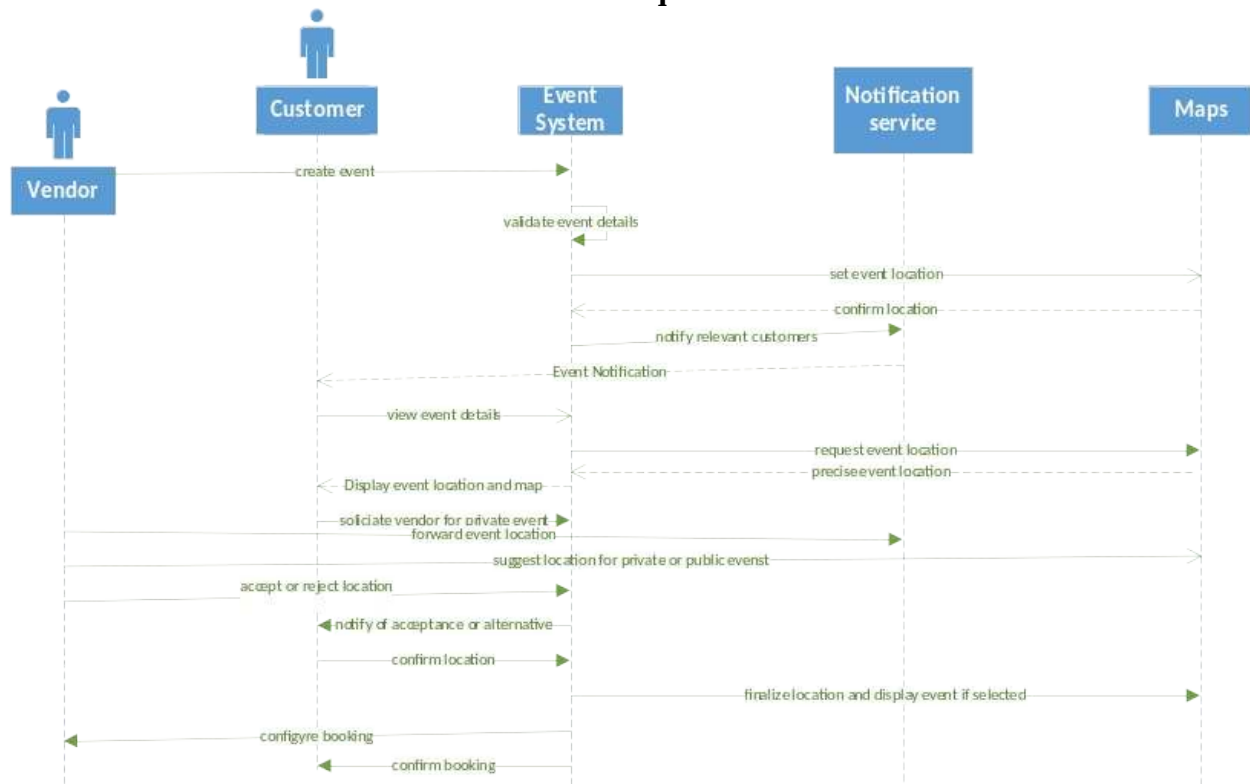


2.4 Sequence Diagrams

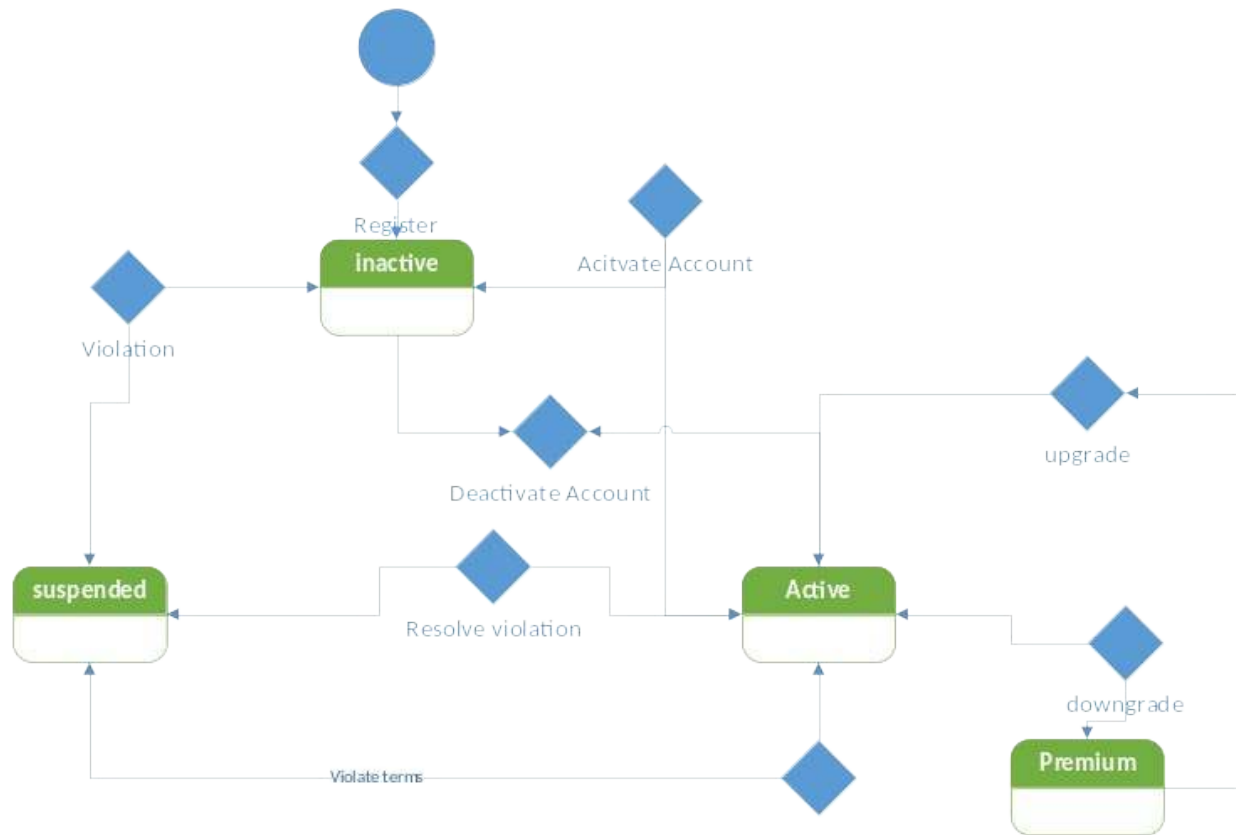
Map Sequence



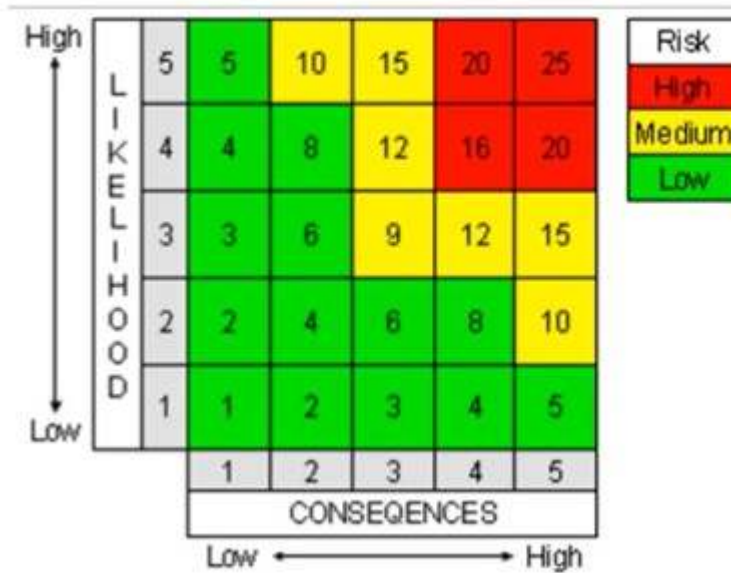
Events Sequence



2.5 State Machine Diagram

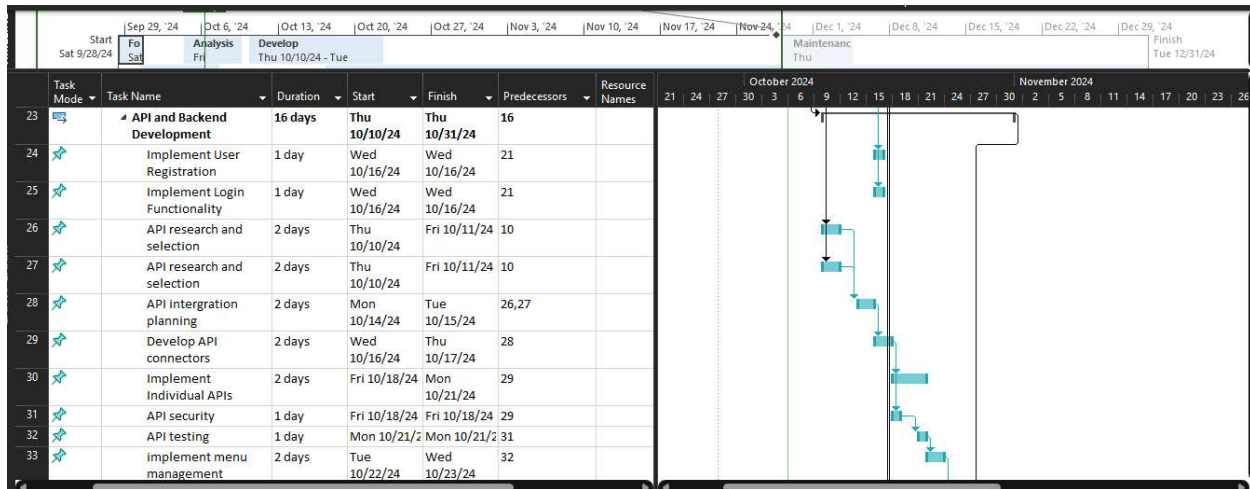


3 Risk Assessment Matrix



Risk ID	Risk Description	Likelihood (1-5)	Consequences (1-5)	Risk Score	Risk Level
R1	Low adoption rate by food truck vendors	4	5	20	High
R2	Data security breach	2	5	10	Medium
R3	System downtime affecting real-time features	2	4	8	Medium
R4	Inaccurate location tracking	3	3	9	Medium
R5	Negative reviews affecting vendor participation	3	3	9	Medium
R6	Scalability issues during peak usage	3	4	12	Medium
R7	Regulatory compliance issues (food safety, privacy laws)	2	4	8	Medium
R8	Competition from similar platforms	3	3	9	Medium
R9	Payment processing failures	2	4	8	Medium
R10	Poor user experience leading to customer churn	3	5	15	High

4 Updated Project Plan



5 Test Plan

5.1 Test Plan Identifier

TP-FTCP-2024-01 (Food Truck Community Platform Test Plan 2024, Version 01)

5.2 Introduction(scope):

The scope of this test plan is to outline a workable strategy for the Food Truck Community Platform. The intent of the plan is to validate key features of the proposed platform. The plan will also verify that specification requirements function web. Mobil, and desktop interfaces.

Because of the need for consistency throughout the testing process a standardized format will be implemented. This means each test case will adhere to a standardized format containing generic elements which will be apparent later in this document. The goal of this standardization is to provide clarity traceability and efficiency to each test to maximize the potential for risk mitigation.

5.3 Test Items

- 6 **Web applications** across devices to test the responsive web design.
- 7 **Mobil applications** for iOS and Android.
- 8 **Backend API services**
- 9 **Database Operations**

10 **Third party integration** such as payment gateways and mapping services.

10.1 Features to be Tested

1. **User Authentication:** User registration and Authentication
2. **Profile Management:** Profile creation and management
3. **Location Tracking:** Real time location tracking
4. **Event Management:** Event creation and management
5. **Feedback System:** Review and rating system
6. **Discovery Features:** Search and discovery features

10.2 Features not to be Tested

Third party libraries and frameworks

10.3 Approach

1. **Unit testing:** to test individual components.
2. **Integration testing:** For API backend and other interactions
3. **User interface testing:** For the web and mobile platforms
4. **Performance testing:** real time location
5. **Security testing:** user data and authentication

10.4 Items Pass/Fail Criteria

The software must pass all the following test cases to be acceptable

1. **Functionality**
Pass: The feature performs its intended function without errors.
Fail: The feature doesn't work as expected or produces errors.
2. **Efficiency**
Pass: The system responds within 2 seconds under normal load.
Fail: Response time exceeds 2 seconds or system becomes unresponsive.
3. **Data Integrity:**
Pass: All data is saved correctly and can be retrieved without corruption.

Fail: Data is lost, corrupted, or cannot be retrieved accurately.

4. **Compatibility:**

Pass: The software works as expected on all specified browsers/devices.

Fail: The software doesn't function properly on one or more specified platforms.

10.5 Test Deliverables

1. **Test plan:** outlines the testing strategy, scope, resources, schedule, and activities
2. **Test cases:** instructions on how to perform testing
3. **Test scripts:** Written scripts to automate certain tasks or make it easier for testers to perform repeated test cases
4. **Test data:** A production-like set of data test cases used to determine whether an application is working correctly
5. **Test report:** A document that contains a summary of all test activities and final test results of testing project
6. **Defect report:** used to integrate into test results

10.6 Environmental Needs

This is a concise list of the basic environmental needs for the test plan but this is not exhaustive and needs can be added in updated versions:

1. **Hardware:**
 - a) Various iOS and Android smartphones and tablets
 - b) Desktop/laptop computers with different specifications
2. **Software:**
 - a) Web browsers
 - b) Latest versions of iOS and Android
 - c) Azure cloud environment for backend
3. **Network:**
 - a) network conditions such as Wi-Fi, 4G, 5G
 - b) Simulated low-bandwidth and high-latency scenarios, AKA throttling
4. **Database:**
 - a) Azure Cosmos DB for NoSQL database operations
5. **Third-party integrations:**
 - a) Payment gateways
 - b) Mapping services such as Google Maps API for Android and Apple Maps API for iOS
 - c) Push notification services

6. **Security:**
 - a) Test environment with security measures like encryption and multi-factor authentication
7. **Localization:**
 - a) Multiple language settings to test multi-language support
8. **Accessibility:**
 - a) Screen readers and other assistive technologies

10.7 Responsibilities

1. **Test Manager:**
 - a) Maintain test strategy
 - b) Coordinate testing
 - c) Report test progress to stakeholders
2. **QA Engineers:**
 - a) Design and review test cases
 - b) Execute complex test scenarios
 - c) Analyze and report test results
3. **QA Testers:**
 - a) Perform manual testing across platforms
 - b) Report and track bugs
 - c) Conduct regression testing
4. **Performance Tester:**
 - a) Design and execute load and stress tests
 - b) Analyze system performance under various conditions
 - c) Identify and report performance bottlenecks
5. **Development Team:**
 - a) Provide timely bug fixes
 - b) Assist in troubleshooting complex issues
 - c) Implement necessary changes based on test results
6. **Project Manager:**
 - a) Ensure testing aligns with project timeline
 - b) Allocate resources for testing activities
 - c) Facilitate communication between testing and other teams

10.8 Staffing and Training Needs

1. **Test Team:**
 - a) 1 Test Manager
 - b) 2 Senior QA Engineers
 - c) 3 QA Testers
 - d) 1 Performance Tester
2. **Required Skills:**
 - a) Mobile and web app testing

- b) API and security testing
- c) Usability and localization testing
- 3. **Training:**
 - a) Platform-specific Azure, Blazor, MAUI
 - b) Test automation and performance tools
 - c) Mobile testing techniques
- 4. **External Resources:**
 - a) Security audit firm
- 5. **Key Responsibilities:**
 - a) Test planning and strategy
 - b) Test case design and execution
 - c) Performance testing
 - d) Bug reporting and regression testing

10.9 Schedule

- 1. **Test Planning:** 2 weeks
 - a) Develop test strategy
 - b) Prepare test cases
- 2. **Environment Setup:** 1 week
- 3. **Unit Testing:** Ongoing
- 4. **Integration Testing:** 3 weeks
- 5. **System Testing:** 4 weeks
 - a) Functional testing
 - b) Non-functional testing
- 6. **User Acceptance Testing:** 2 weeks
- 8. **Bug Fixes and Final Testing:** 2 weeks

10.10 Risks and contingencies

- 1. **Resource Shortage:** Short staffed with proper skillset.
 - a) **Contingency:** Cross train staff to test as they develop\
 - b) **Contingency:** Hire temporary outside contractors
- 2. **Timeline Pressure:** Rushed testing due to delays.
 - a) **Contingency:** Attach priority metric to features.

3. **Test Data:** Lack thereof or not reliable.
 - a) **Contingency:** Use anonymized data from similar projects if available.
 - b) **Contingency:** Create sample data.
4. **Tool Reliability:** Testing tools producing inconsistent or unreliable results.
 - a) **Contingency:** Rotate between multiple testing tools to cross-validate results
5. **Communication Gap:** Miscommunications and delays in communication between dev and test teams.
 - a) **Contingency:** Have daily formal review meetings with from members of both teams.
 - b) **Contingency:** Enforce the use of collaborative tools.
6. **Environmental Issues:** Testing environments are unstable.
 - a) **Contingency:** Back up environments.
 - b) **Contingency:** Back up testing locations.
7. **Scope Creep:** Testing leads to finding errors and bugs which leads to changing features or reworking them. This can change your timeline.
 - a) **Contingency:** Change Control Process
 - b) **Contingency:** Consistently reassess priorities.

2

11 Test Cases

11.1 Test Cases

Test Case #1

Test Case ID: TC-UA-001

Feature: User Authentication

Objective: Verify that a user can successfully register and log in to the platform.

Preconditions:

1. The application is accessible
2. Test user email is not already registered

Test Steps:

1. Navigate to the registration page
2. Enter valid user information (name, email, password)
3. Submit the registration form
4. Check email for confirmation
5. Click on the confirmation link in the email
6. Navigate to the login page
7. Enter the registered email and password
8. Submit the login form

Expected Results:

1. User registration is successful
2. Confirmation email is received
3. User can log in with registered credentials
4. After login, user is directed to their dashboard

Pass/Fail Criteria:

1. Pass: All expected results are achieved
2. Fail: Any expected result is not achieved

Actual Results: [To be filled during test execution]

Status: [To be filled after test execution]

Notes: [Any additional observations]

Test Case #2

Test Case ID: TC-UL-001

Feature: Real-time Location Tracking

Objective: Verify that a vendor's location is accurately tracked and displayed on the map

Preconditions:

1. Vendor account is created and verified
2. Customer account is created and verified

3. Vendor has location tracking enabled

Test Steps:

1. Log in as a vendor
2. Enable location tracking
3. Move to different locations (simulate if testing in a controlled environment)
4. Log in as a customer
5. Search for the vendor
6. View the vendor's location on the map

Expected Results:

1. Vendor's location updates in real-time
2. Customer can see the vendor's current location on the map
3. Location updates are within 50 meters of accuracy
4. Location refreshes at least every 60 seconds

Pass/Fail Criteria:

1. Pass: All expected results are achieved
2. Fail: Any expected result is not achieved

Actual Results: [To be filled during test execution]

Status: [To be filled after test execution]

Notes: [Any additional observations]

Test Case #3

Test Case ID: TC-EM-001

Feature: Event Creation Management

Objective: Verify that a vendor can create an event and customers can discover and join it

Preconditions:

1. Vendor account is created and verified
2. Customer account is created and verified

Test Steps:

1. Log in as a vendor
2. Navigate to the event creation page
3. Fill in event details (name, date, time, location, description)
4. Submit the event
5. Log in as a customer
6. Use the search feature to find the created event
7. View event details
8. Join the event

Expected Results:

1. Vendor can create an event successfully
2. Event appears in search results for customers
3. Customers can view all event details
4. Customers can join the event
5. Vendor receives a notification when a customer joins their event

Pass/Fail Criteria:

1. Pass: All expected results are achieved
2. Fail: Any expected result is not achieved

Actual Results: [To be filled during test execution]

Status: [To be filled after test execution]

Notes: [Any additional observations]

12 Updated Project Plan

<Provide updated WBS Project Plan screen capture>

Appendix A: Glossary

API application programming interface: Rules, protocols and tools for allowing different types of software to communicate and work together they are used when developing software.

Authentication: Verification of a user or system

Azure: Microsoft's cloud computing platform

Blazor: A web framework for building interactive web UIs using the .NET framework and C# and other languages.

Customer/Enthusiast: Users of the platform that are not vendors.

Entity Framework Core: an object relational mapping for .NET

Food truck/stand: All type of food vendors.

IEEE 830: A standard for SRS documents.

Maui multi platform map UI: .NET open source framework for building native mobile and desktop apps with C# and XAML.

NoSQL: Not Only SQL stores distributed data with humongous storage needs they are very elastic.

OAuth 2.0: industry standard protocol for authorization

OpenID Connect: Identity layer built on top of OAuth 2.0 protocol.

Push Notifications: A message that pops up on your device

REST representational state transfer: An architectural design for designing network applications.

SignalR: Software library for Microsoft ASP.NET that allows server code to send asynchronous notifications to client-side web applications

Vendor: Food truck/stand owners that use the platform to endorse business.

WebSocket: Communication protocol, allows two way communication over a single tcp connection, this is a vital part of the Blazor server.

Appendix 1 Use Case Types

Business Use Cases:

- Describe high-level business processes
- Focus on business goals and outcomes
- Usually written in business language, not technical terms

System Use Cases:

- Describe interactions between users and a specific system
- More detailed than business use cases

- Focus on system functionality and user interactions

Abstract Use Cases:

- Generalized use cases that can be reused across multiple scenarios
- Often used as a template for more specific use cases

Concrete Use Cases:

- Specific instances or implementations of abstract use cases
- Provide detailed descriptions of particular scenarios

Primary Use Cases:

- Represent the main or most common scenarios
- Essential for achieving the primary goals of the system

Secondary Use Cases:

- Represent less frequent or optional scenarios
- Support the primary use cases but are not essential

Brief Use Cases:

- Concise, high-level descriptions
- Usually only a few sentences long
- Provide an overview without much detail

Casual Use Cases:

- More detailed than brief use cases
- Include main success scenario and some alternative flows
- Less formal and structured than fully dressed use cases

Fully Dressed Use Cases:

- Most detailed and comprehensive type
- Include all scenarios, preconditions, postconditions, and other elements
- Typically used for critical or complex functionalities

Appendix 2 Importance level

1. Essential (Critical / Core)
 - o Absolutely necessary for the system to function
 - o Must be implemented for the system to be considered complete
 - o Directly supports primary business objectives
2. Important (High)
 - o Significant impact on system functionality
 - o Should be implemented, but the system can function without them
 - o Provides substantial value to users or the business
3. Desirable (Medium)
 - o Adds value but is not crucial for core functionality
 - o Can be deferred to later iterations or releases if necessary
 - o Enhances user experience or provides additional features
4. Optional (Low)
 - o Nice-to-have features
 - o Can be easily omitted without significantly impacting the system
 - o May be implemented if time and resources allow
5. Future Consideration

- o Not planned for the current version or release
- o Potential features for future enhancements
- o Kept in the backlog for later evaluation