

CEIS 200

Software Enigneering 1

Louie Venegas

Specification Requirements Project Level Summary

What would you do better or differently?

The document needs consistent heading numbering (particularly in sections 2.5 through 2.5.4) and proper formatting for IEEE 830 compliance. The appendices should include more detailed diagrams. The project scope could benefit from clearer prioritization of features and more specific technical requirements for the mobile application.

What was the quality of the tools and techniques?

The selection of modern development tools and frameworks (.NET 6.0, Blazor, MAUI, Azure) is appropriate for the project's needs and demonstrates good technical planning. The use of OAuth 2.0 and OpenID Connect for authentication shows attention to security standards, though more specific details about implementation techniques would strengthen the document.

What are the strengths and weaknesses?

The document's main strength is its comprehensive coverage of both business and technical aspects, particularly in defining user classes and external interfaces. The primary weakness lies in undefined success metrics for features, missing implementation details for the loyalty program, and incomplete security requirements particularly around data protection and compliance standards.

What are your recommendations or suggestions?

The document would benefit from including detailed API specifications, adding concrete performance benchmarks, and providing more specific security requirements especially around data protection regulations. The project plan section needs expansion with clear milestones, and the use case diagrams should be properly embedded with higher resolution.

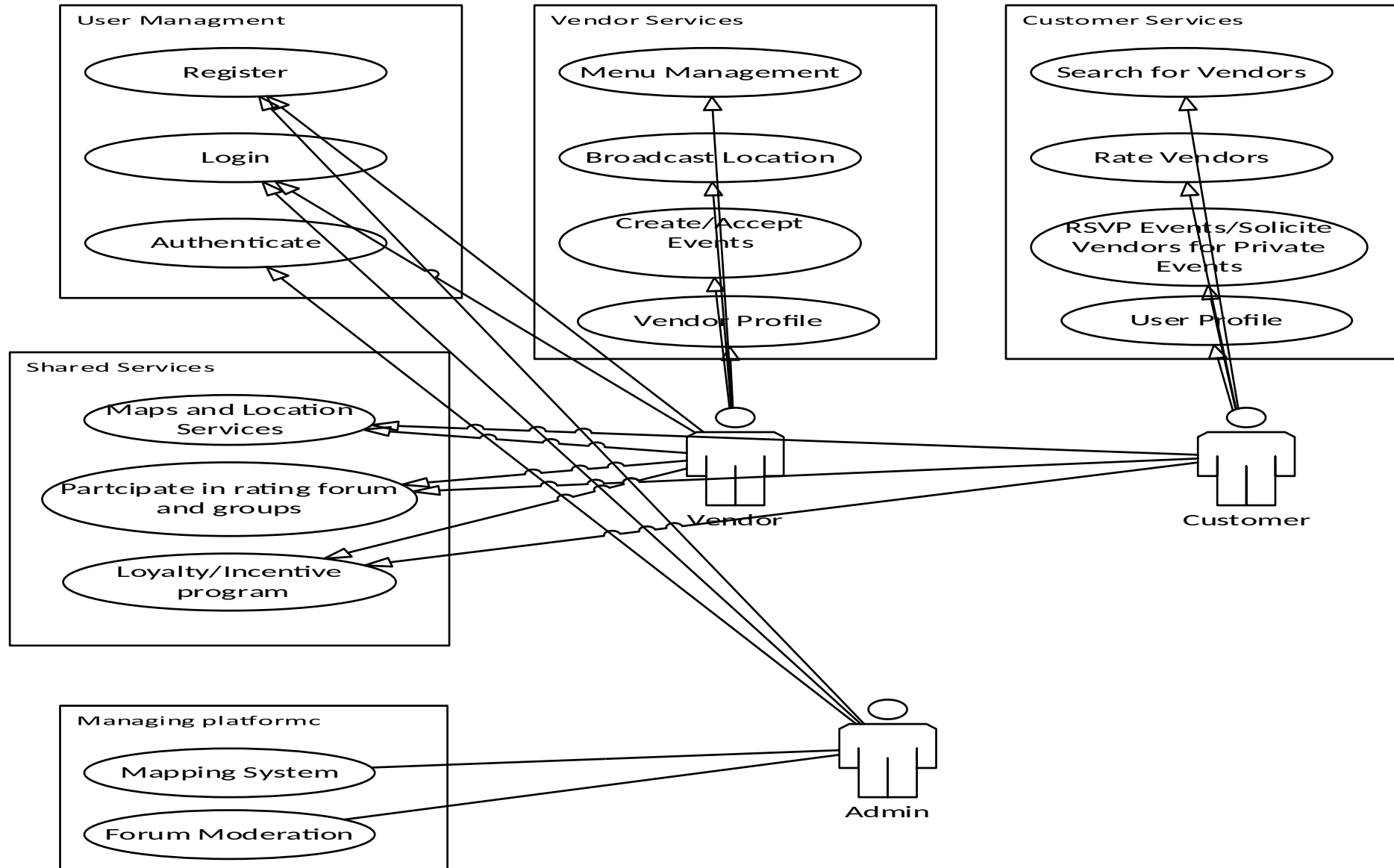
Use Case Diagram

In the following slide on display is a use case diagram pulled from one of the work products produced. This use case diagram is showing the functionality and user interactions of a system. The diagram is broken down into several key sections: The diagram shows three main types of users (represented by stick figures), Regular Users, Vendors and Admin. Each line connecting the users to the functions (ovals) represents what actions or features each type of user can access in the system.

This type of diagram is commonly used in software engineering and system design to visualize:

- What features are available in the system
- Who can access which features
- How different parts of the system are organized
- The relationships between users and functionality.

Use Case Diagram



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¶		

1.2.7-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		
Sub-Flows:- 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- -to-enter-a-location-manually Proceeds-with-search-based-on-entered-location		

Project Level Summary – Analysis and Design

What would you do better or differently?

The document's use case specifications are thorough but could benefit from clearer formatting and consistent numbering throughout the sections. Additionally, the risk assessment matrix could include mitigation strategies for each identified risk. The loyalty program specifications need more detailed success metrics and concrete reward structures. The event creation process could be streamlined by including templates for common event types and automated vendor matching algorithms.

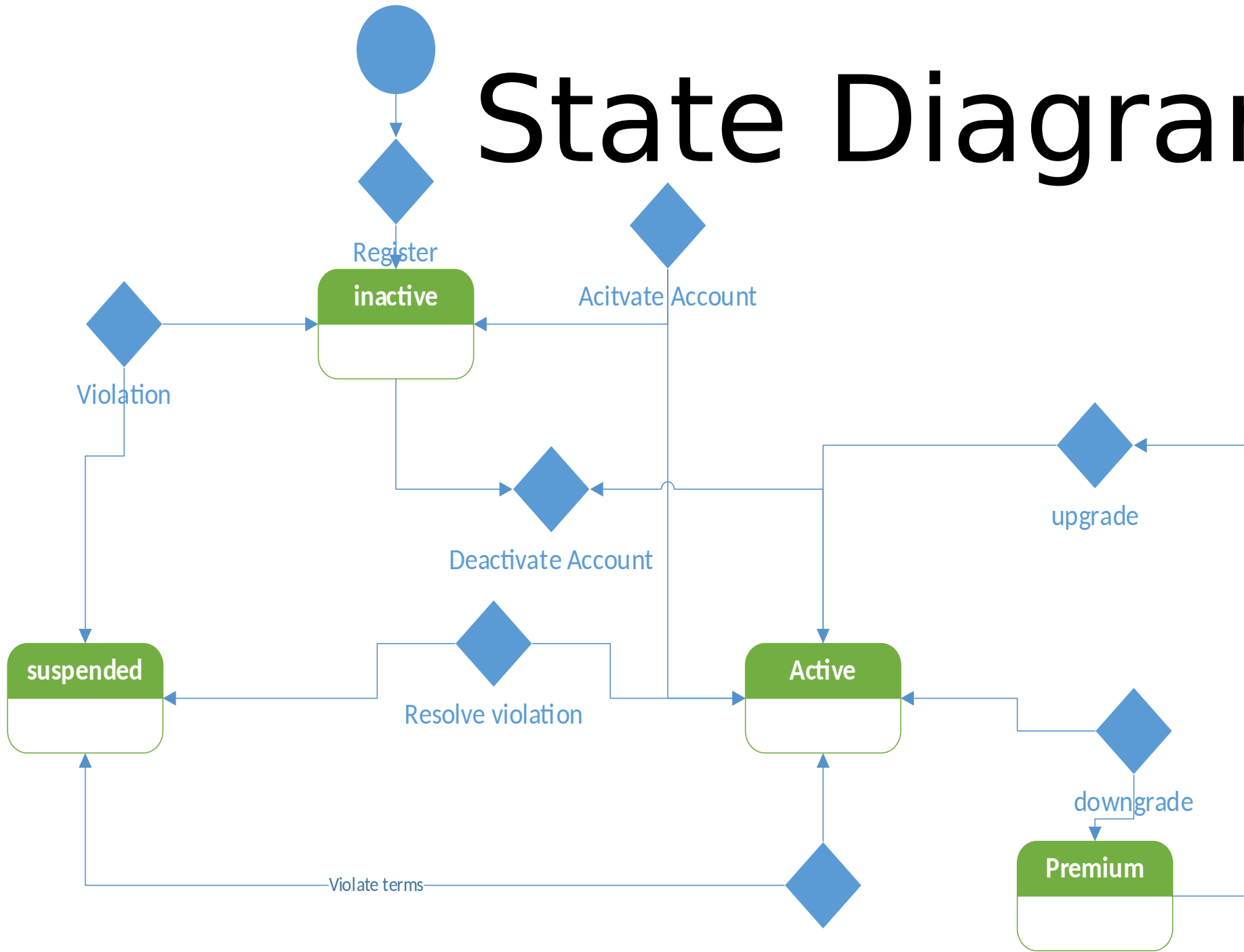
Do they have a good understanding of key risks for their projects and how to mitigate risk (external and internal)?

The risk assessment matrix demonstrates a solid understanding of both technical and business risks, identifying key concerns like vendor adoption rates and data security. However, while risks are well identified and scored, the document lacks explicit mitigation strategies and contingency plans for each risk, which would strengthen the risk management approach. The project would benefit from a more detailed analysis of competitive risks and market penetration strategies. The internal risks related to development team capabilities and resource allocation should also be more thoroughly addressed.

What additional risks would you consider for the project (minimum of three)?

Third-party API dependency risks related to mapping services failure, event coordination risks during high-traffic periods like festivals or holidays, and vendor verification risks related to food safety compliance and liability issues should be considered. These could significantly impact platform reliability and user trust. The potential risk of fraudulent vendor accounts and fake reviews needs to be addressed with robust verification systems and community moderation tools. There's also a significant risk related to seasonal business fluctuations and maintaining vendor engagement during slow periods.-

State Diagram

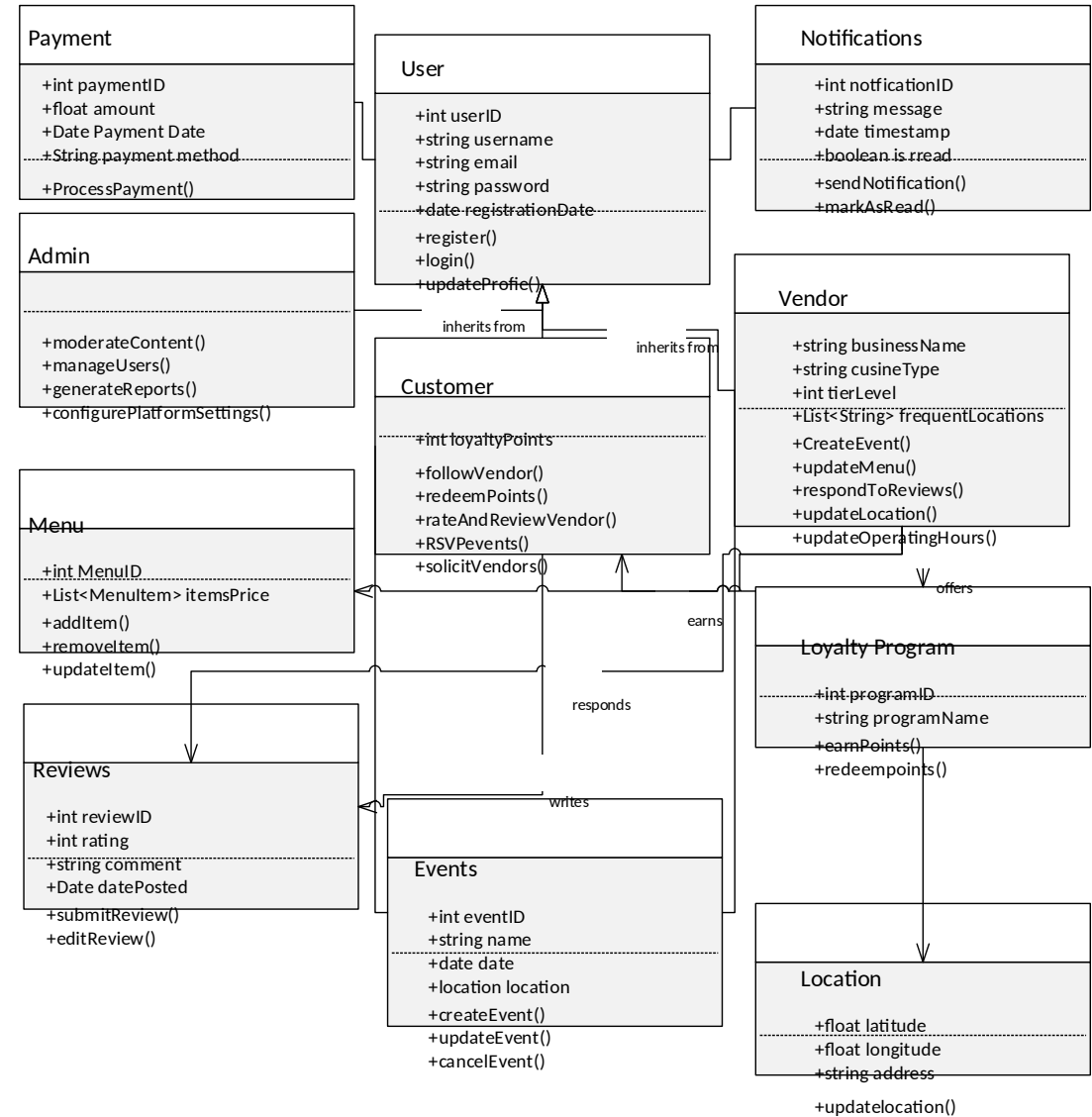


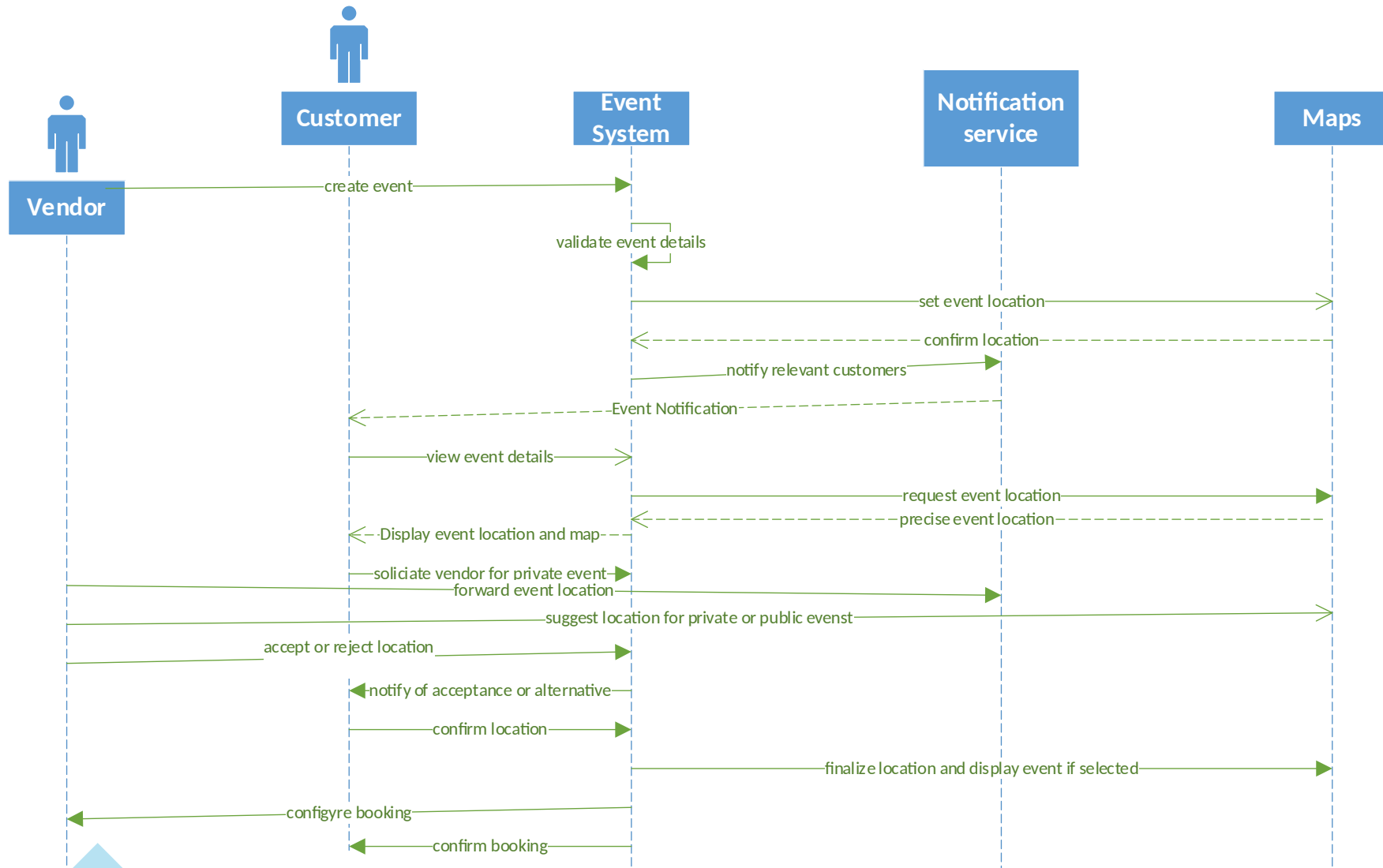
This type of diagram is commonly used in software development and system design to illustrate how an account or user management system works, showing all possible states an account can be in and how it can transition between those states. It helps developers and stakeholders

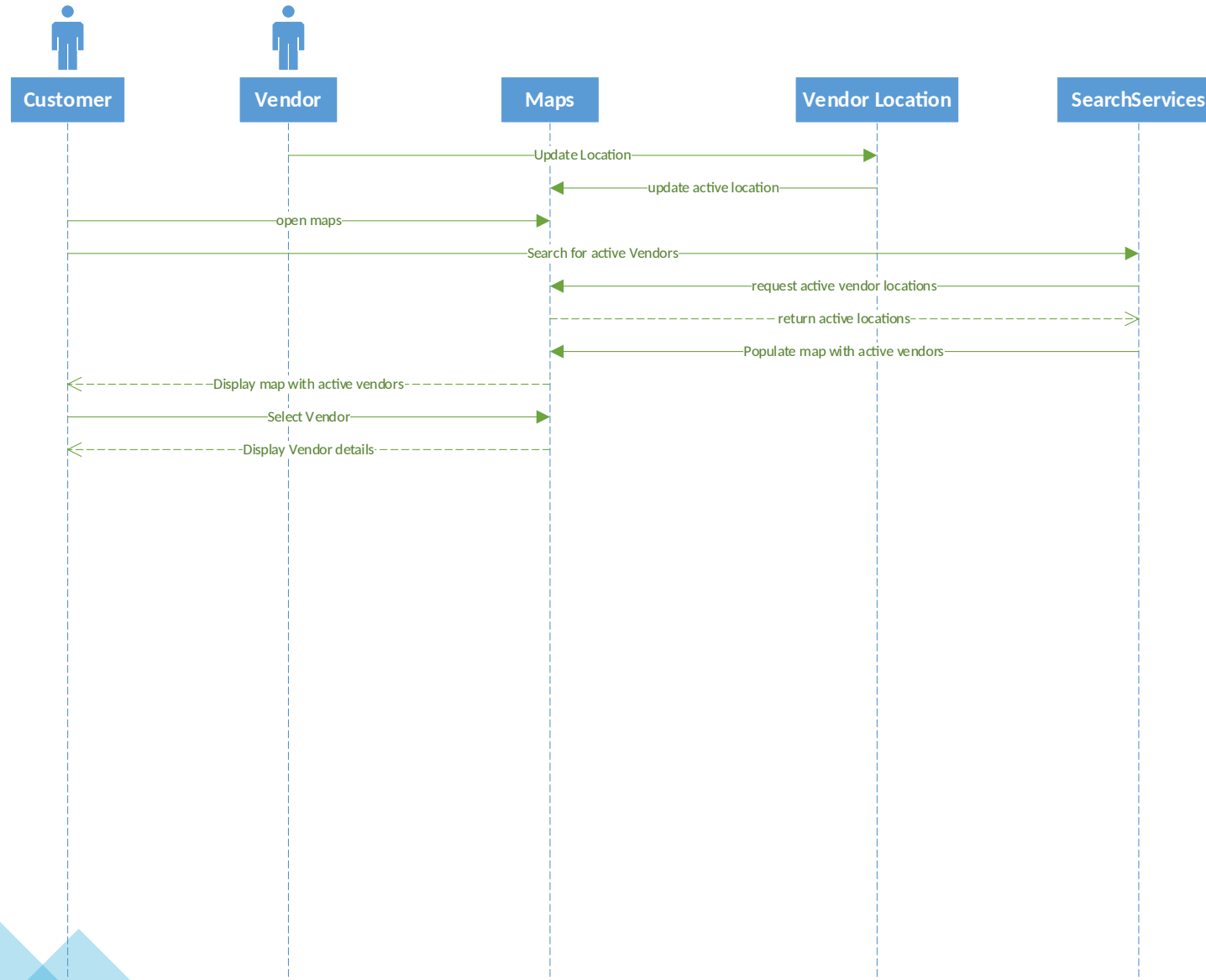
UML Class Diagram

•The following slide shows a UML Class Diagram, a tool used in software engineering for the illustrating the structure of a system. The following diagram maps out various classes, their attributes, methods, and the relationships between different classes. This class diagram is organized around several key components, such as user management, payment processing, and menu functionality.

•Each class in the diagram is in a box containing its attributes and methods. The lines connecting these boxes illustrates the relationships between different classes, creating a map of how the components work together. This diagram is used for object-oriented design providing the developers with a clear visualization of the system's architecture helping them understand the way in which parts of the system interact with each other.







Project Level Summary Evaluation – Testing

What would you do better or differently?

The test plan lacks detailed performance benchmarks and specific metrics particularly for real-time features like location tracking. The test cases should include more edge cases and negative testing scenarios to ensure error handling. The security testing section needs expansion, especially regarding data privacy and third-party integration vulnerabilities.

Do you get a good understanding of their testing process or the thoroughness of their testing? Yes or no? Explain either way.

The testing process is well-structured but lacks depth in certain critical areas such as security testing and performance benchmarks. While the basic testing framework is established, more detailed test scenarios for complex user interactions and system integrations are needed. The test schedule appears optimistic given the complexity of features to be tested and potential resource constraints.

Is there traceability from requirements?

The test cases show clear links to major functional requirements but lack explicit traceability to non-functional requirements and business rules. The plan should include a traceability matrix mapping each test case to specific requirements and user stories. More comprehensive coverage of edge cases and integration points would strengthen requirement validation.

Do they have a test plan and test cases? If so, describe the quality. How do they apply to the three testing concepts of verify, validate, and explore?

The test plan and cases are present but need more comprehensive coverage of negative testing scenarios and performance thresholds. While verification of functional requirements is well-covered, validation of user experience and exploratory testing need more attention. The test cases focus heavily on verification but should expand validation and exploratory testing aspects.

How would your new functional requirements for the project impact the testing process?

The testing process would need significant expansion to cover new features and their interactions with existing functionality. Additional test environments and data sets would be required to properly validate new requirements. Test schedules and resource allocations would need adjustment to accommodate the increased scope.

Overview of the Test Plan

Main Testing Areas:

- User accounts (signing up, logging in)
- Food truck location tracking
- Event management
- Customer reviews and ratings
- Search features

Testing Methods:

- Testing individual parts of the code
- Making sure everything works together
- Checking the user interface on phones and computers
- Testing how well the app performs under heavy use
- Making sure user data is secure

Things We Need:

- Different phones and computers for testing
- Internet connections (WiFi, 4G, 5G)
- Test payment systems
- Map services for location tracking

Team:

- 1 test manager
- 2 senior testers
- 3 regular testers
- 1 performance specialist

Timeline: We're planning about 14 weeks total, breaking down into:

- 2 weeks for planning
- 8 weeks for main testing
- 2 weeks for user testing
- 2 weeks for fixing problems

Main Concerns:

- Having enough people to test everything
- Keeping to our schedule
- Making sure our test data is good
- Keeping test environments stable

Test Case

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:

3. Navigate to the registration page
4. Enter valid user information (name, email, password)
5. Submit the registration form
6. Check email for confirmation
7. Click on the confirmation link in the email
8. Navigate to the login page
9. Enter the registered email and password
10. 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:

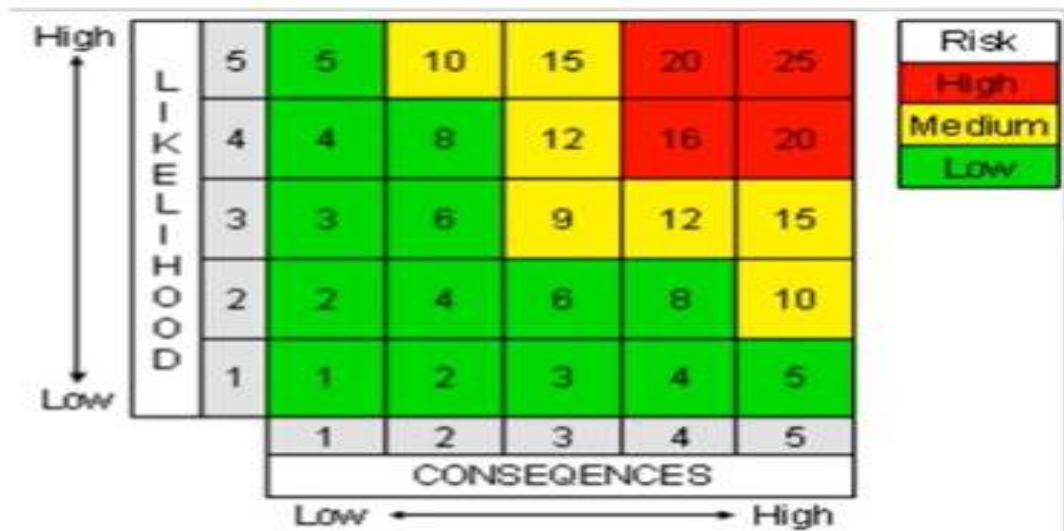
5. Pass: All expected results are achieved
6. Fail: Any expected result is not achieved

Actual Results: [To be filled during test execution]

Status: [To be filled after test execution]

Notes: [Any additional observations]

1 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

Summarize what you have learned based on the software engineering concepts and applications defined throughout this course.

Throughout the course of this class I learned about various topics in software engineering. The principles of developing UML models that represent structural and behavioral aspects of software systems were introduced and put into practice. A concept that received a lot of emphasis was designing software from requirements specifications as well as developing those specifications. By following the requirements specifications laid out in this course I can ensure portability, usability, high performance and high-quality software. The best practices were laid out in detail for risk analysis and assessment. The quantification of risk analysis and other software engineering methods were introduced and provided me with a way to objectively look at numbers that define the project and see both the progress and efficiency. Testing was another area that received quite a bit of detail this is appropriate because testing is what ensures the software product is at its greatest potential and as free of bugs as possible. Overall, as software project management was a major theme in this course a large part of this course was dedicated to developing the work products that surround the software development lifecycle.

The principles and concepts taught in this class will lay down a solid foundation upon which to develop a career in software development or project management. Understanding the work products and the reasons they exist will go a long way in the field since the industry flows on the back of these products. Being introduced Software process models, agile methodologies, people management as well as project management concepts will help me during an interview. While I can not claim to be an expert on these things the base knowledge I have will provide an employer with enough foundation to build a valuable employee. Overall, the knowledge acquired in this course will prove invaluable overtime and help me pursuing and achieving my goals and ambitions.

Tools I Recommend for Our Product:

Visual Studio 2022 - It's like a smart notepad for writing our app's code. It's made by Microsoft and works well with the other tools we want to use.

Blazor - This helps us make the website part of our app. It uses a language called C# which is easier for beginners.

.NET MAUI - This is for making the phone app. It lets us use the same code for iPhones and Androids.

Azure DevOps - This helps us keep track of our work and make sure everything's organized.

Azure Cosmos DB - This is where we'll store all our app's information. It works well with the other Microsoft tools we're using.

I chose these because they're all part of the Microsoft family, so they should work well together. They're also good for people who are just starting to make apps.