Software Requirements Specification

for

<Project>

Version <X.X>

Prepared by

Group Name: ***<place your group name here>***

<name>	**<student #>**	**<e-mail>**
<name>	**<student #>**	**<e-mail>**
<name>	**<student #>**	**<e-mail>**
<name>	**<student #>**	**<e-mail>**
<name>	**<student #>**	**<e-mail>**

Instructor:	***<place your instructor's name here>***
Course:	**<place your course name here>**
Lab Section:	***<place your lab section here>***
Teaching Assistant:	***<place your TA's name here>***
Date:	**<place the date of submission here>**

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

< The purpose of the "FoodXpress" project is to provide a convenient and efficient solution for students to order meals in advance from their university mess, canteen, or nearby restaurants. This native Android application aims to streamline the food ordering process, allowing customers to place their orders remotely and make payments using the Razorpay API seamlessly. By leveraging Firebase for database management, the app ensures real-time updates and order tracking. The primary goal is to enhance the overall dining experience for students by reducing waiting times and optimizing food service in educational institutions. "FoodXpress" aims to revolutionize how students interact with dining facilities, offering a faster and more convenient way to enjoy meals on campus.>

1.1 Document Purpose

< In the Software Requirements Specification (SRS) document for the "FoodXpress" project, the purpose section should succinctly outline the objectives and goals of the document. Here's a suggested statement for the purpose section:

"The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the Mess Preordering System within the FoodXpress project. It aims to provide a comprehensive understanding of the software's scope, features, and interactions, ensuring a clear roadmap for development and successful implementation. This document serves as a primary reference for stakeholders, including developers, testers, and project managers, to align their efforts and deliver a high-quality preordering system that meets the needs of users and stakeholders.".>

1.2 Product Scope

< The project scope for FoodXpress encompasses the development of a native Android application designed specifically for students. The primary goal of the application is to streamline the food ordering process in various dining establishments such as messes, restaurants, and canteens within educational institutions. Users will have the capability to place orders in advance through the app, enabling them to avoid long wait times upon arrival at the dining venue. The scope includes integrating third-party APIs such as Razorpay for secure and convenient payment processing. Additionally, the application will leverage Firebase for efficient data management and storage, ensuring seamless order management and user experience. Overall, the project aims to enhance customer satisfaction by providing a faster and more convenient dining experience for students.>

1.3 Intended Audience and Document Overview

< *The intended audience for this SRS document includes project stakeholders, developers, testers, project managers, and anyone involved in the development and maintenance of the FoodXpress application.*

For project stakeholders, such as clients, investors, and business analysts, this document serves as a comprehensive overview of the project's scope, objectives, functionalities, and requirements. It provides insights into the expected features of the FoodXpress application and outlines the technical aspects that will be implemented.

Developers and testers will find detailed information about the system architecture, design specifications, use cases, and functional requirements in this document. It serves as a reference guide for understanding the system's behavior, interactions, and data flow, aiding in the development, testing, and debugging phases.

Project managers can use this document to track progress, manage expectations, and ensure that the development team is aligned with the project goals ad requirements. It helps in scheduling tasks, allocating resources, and prioritizing features based on business needs and user requirements.

To effectively navigate this document, readers are encouraged to start with the Introduction section to gain a high-level understanding of the project's purpose, followed by the Requirements section for detailed functional and non-functional requirements. The Use Cases section provides specific scenarios and interactions, while the Design and Architecture sections delve into technical specifications and system components. Additionally, the Appendices contain supplementary information, such as glossaries, acronyms, and references, for further clarification.

By following this structured approach, readers can grasp the essence of the FoodXpress project, understand their respective roles and responsibilities, and contribute effectively to its successful development and deployment.>

1.4 Definitions, Acronyms and Abbreviations

<In the "Definitions, Acronyms and Abbreviations" section of the SRS document for the "FoodXpress" project, you should include definitions for any technical terms, acronyms, or abbreviations used throughout the document to ensure clarity and consistency. Here's an example of what this section might include:

- **Mess Preordering System**: Refers to the software component of the FoodXpress project that allows users to place food orders in advance from a designated dining facility, typically within a military or institutional setting.
- **SRS**: Software Requirements Specification.
- **UI**: User Interface.
- **API**: Application Programming Interface.
- **POS**: Point of Sale.
- **DBMS**: Database Management System.
- **HTTP**: Hypertext Transfer Protocol.
- **GUI**: Graphical User Interface.
- **UX**: User Experience.
- **CSV**: Comma-Separated Values.
- **QA**: Quality Assurance.
- **UAT**: User Acceptance Testing.
- **PCI DSS**: Payment Card Industry Data Security Standard.

Include any other terms specific to your project or industry that may require clarification for stakeholders or team members reviewing the document.>

1.5 Document Conventions

<1. **Naming Conventions:**
   - Use camelCase for naming variables, functions, and methods in Java.

- XML files should follow a consistent naming convention, such as lowercase with underscores for readability.
- Class names should be in PascalCase to adhere to Java naming conventions.

2. **Coding Standards:**
   - Follow Java coding standards and best practices for writing clean, readable, and maintainable code.
   - Use meaningful variable names and comments to improve code understanding.
   - Ensure proper indentation and formatting for XML files to enhance readability.

3. **API Integration Guidelines:**
   - Follow the guidelines provided by third-party APIs, such as Razorpay for payment integration.
   - Handle API responses and errors gracefully to provide a smooth user experience.
   - Maintain security standards when dealing with sensitive data, especially during payment transactions.

4. **Database Naming Conventions:**
   - Use meaningful names for database tables, fields, and collections in Firebase.
   - Follow a consistent naming convention to organize and manage data efficiently.
   - Document the database schema and relationships for clarity and future reference.

5. **Version Control Conventions:**
   - Use version control systems like Git to track changes and collaborate on the project.
   - Follow branching and merging strategies to manage codebase versions effectively.
   - Document version history and release notes for transparency and accountability.

6. **Documentation Format:**
   - Use Markdown or a similar markup language for documenting project-related information.
   - Include sections such as Introduction, Requirements, Design, Implementation, Testing, etc., in the SRS document.
   - Provide clear headings, subheadings, and table of contents for easy navigation within the document.

These conventions ensure consistency, clarity, and maintainability throughout the development process of the "FoodXpress" application.>

1.6 References and Acknowledgments

<*List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.* >

2 Overall Description

2.1 Product Overview

<
Product Overview

The mess preordering system, a vital component of the FoodXpress project, is designed to streamline the food ordering process for users within institutional dining facilities, such as military mess halls or corporate cafeterias. FoodXpress aims to modernize and enhance the dining experience by allowing patrons to conveniently place and manage food orders in advance, reducing wait times and improving overall efficiency.

Key Features:

- **Menu Browsing and Selection:** Users can browse through a comprehensive menu of available dishes and beverages, selecting their desired items for preordering.

- **Customization Options:** The system allows users to customize their orders according to dietary preferences, portion sizes, and other specifications.

- **Order Management:** Users can view and manage their current and past orders, including the ability to modify or cancel orders before the designated cutoff time.

- **Integration with Payment Systems:** Seamless integration with payment processing systems enables users to securely complete transactions for their preorders.

- **Notification and Confirmation:** Users receive timely notifications and confirmations regarding the status of their orders, including pickup times and location information.

Scope:

The mess preordering system covers the entire ordering process, from menu browsing to order fulfillment, within institutional dining facilities participating in the FoodXpress project. It interfaces with existing dining management systems and integrates seamlessly with backend databases and payment gateways to ensure a cohesive and user-friendly experience.

Tailor this overview to reflect the specific goals, features, and scope of your mess preordering system within the FoodXpress project, providing readers with a clear understanding of the product's purpose and functionality..>

2.2 Product Functionality

<

Product Functionality

The mess preordering system within the FoodXpress project offers a range of functionalities to facilitate seamless food ordering and management for users within institutional dining facilities. The following are key features and capabilities of the system:

1. **User Registration and Authentication:**
   - Users can create accounts or profiles within the system, providing necessary information such as name, email address, and affiliation with the institution.
   - Secure authentication mechanisms ensure that only authorized users can access the system and place orders.

2. **Menu Browsing and Selection:**
   - A user-friendly interface allows patrons to browse through an extensive menu of available dishes and beverages.
   - Intuitive navigation and search functionalities enable users to quickly find and select their desired items for preordering.

3. **Order Placement and Customization:**
   - Users can easily add selected items to their cart and specify customization options such as portion sizes, dietary preferences, and special instructions.
   - The system provides real-time feedback on item availability and order constraints, helping users make informed decisions.

4. **Order Management:**
   - Users have access to a centralized dashboard or interface where they can view and manage their current and past orders.
   - Options for order modification, cancellation, and order history retrieval are available to users for added convenience.

5. **Integration with Payment Systems:**
   - Seamless integration with payment processing systems allows users to securely complete transactions for their preorders.
   - Multiple payment methods, including credit/debit cards and institutional meal plans, are supported to accommodate diverse user preferences.

6. **Notification and Confirmation:**
   - Users receive timely notifications and confirmations via email or SMS regarding the status of their orders.
   - Notifications include order confirmation, pickup/delivery details, and updates on any changes or delays.

7. **Accessibility and Usability:**
   - The system is designed to be accessible and user-friendly, with support for diverse devices and screen sizes.
   - Accessibility features such as screen reader compatibility and keyboard navigation enhance usability for all users.

Tailor this functionality overview to reflect the specific features and capabilities of your mess preordering system within the FoodXpress project, providing a detailed roadmap of the system's functionalities for stakeholders and development teams.
>

2.3 Design and Implementation Constraints

Design and Implementation Constraints

*1. **Technology Stack Compatibility:***
- The mess preordering system must be compatible with the existing technology stack and infrastructure used within institutional dining facilities, including hardware devices, operating systems, and network configurations.

*2. **Integration with Legacy Systems:***
- Integration with legacy systems, such as point-of-sale (POS) systems and inventory management software, may present challenges due to differences in data formats, protocols, and APIs.
- Efforts will be made to ensure seamless integration through the use of standardized interfaces and protocols where possible.

*3. **Data Security and Compliance Requirements:***
- The system must adhere to strict data security and compliance standards, including but not limited to PCI DSS (Payment Card Industry Data Security Standard) for payment processing and GDPR (General Data Protection Regulation) for handling personal data.
- Measures such as data encryption, access controls, and regular security audits will be implemented to mitigate risks and ensure compliance.

*4. **Scalability and Performance Considerations:***
- The system must be designed to accommodate potential scalability requirements, particularly during peak usage periods or when expanding to additional dining facilities.
- Performance optimizations, such as caching, load balancing, and database indexing, will be implemented to ensure optimal system performance under varying workloads.

*5. **User Accessibility and Usability:***
- Accessibility considerations must be taken into account to ensure that the system is usable by individuals with disabilities, including compliance with WCAG (Web Content Accessibility Guidelines) standards.
- User interface (UI) design and interaction patterns will be optimized for usability across different devices and screen sizes.

*6. **Regulatory Compliance and Institutional Policies:***
- The system must comply with relevant institutional policies, regulations, and industry standards governing food service operations, including health and safety regulations and institutional procurement policies.
- Legal and regulatory requirements specific to the jurisdiction(s) where the system will be deployed must be carefully considered and addressed.

Tailor this section to include any specific constraints or challenges relevant to the design and implementation of your mess preordering system within the FoodXpress project, providing a clear understanding of the factors that may impact the development process.. >

2.4 Assumptions and Dependencies

<
Assumptions and Dependencies

1. **Availability of Internet Connectivity:**
   - It is assumed that users of the mess preordering system will have reliable internet connectivity to access the system and place orders. Any disruptions in internet connectivity may impact the user experience and order processing capabilities.

2. **Compatibility with Mobile Devices:**
   - The system is designed to be accessible via web browsers on mobile devices, including smartphones and tablets. It is assumed that users will have access to compatible devices for accessing the system.

3. **Availability of Menu and Inventory Data:**
   - The accuracy and availability of menu and inventory data, including item descriptions, prices, and availability status, are assumed to be maintained by the institutional dining facility's management team or relevant stakeholders.

4. **Integration with Payment Gateway:**
   - Dependency exists on third-party payment gateway services for processing online payments. It is assumed that the payment gateway service will be available and properly configured to facilitate secure transactions.

5. **Timely Notification Delivery:**
   - Assumption is made that notifications regarding order status updates, including order confirmations and pickup/delivery notifications, will be delivered to users in a timely manner via email or SMS.

6. **Compliance with Institutional Policies:**
   - The system's compliance with institutional policies, including food safety regulations, procurement policies, and operational guidelines, is assumed. Any changes to these policies may impact system requirements and functionality.

7. **Availability of Development Resources:**
   - Dependency exists on the availability of development resources, including software engineers, designers, and testers, to implement and test the system within the specified timeline and budget constraints.

Tailor this section to include any specific assumptions or dependencies relevant to the development and operation of your mess preordering system within the FoodXpress project, providing clarity on factors that may influence project outcomes.>

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

User Interfaces

1. **Customer-Facing Interface:**
   - The customer-facing interface serves as the primary interaction point for users placing food orders through the mess preordering system. Key components of the interface include:
     - **Menu Browse and Selection:** A visually appealing and intuitive menu browsing interface allows users to explore available dishes and beverages, view item descriptions, and select items for preordering.
     - **Order Customization:** Users can customize their orders according to preferences such as portion sizes, dietary requirements, and special instructions, using interactive form fields or dropdown menus.
     - **Order Management:** A user-friendly dashboard or order history section enables users to view and manage their current and past orders, including options for order modification, cancellation, and order status tracking.
     - **Checkout and Payment:** Seamless integration with payment processing services provides users with a secure and streamlined checkout experience, supporting various payment methods such as credit/debit cards and institutional meal plans.

2. **Administrative Interface:**
   - The administrative interface is used by system administrators or staff members responsible for managing menu items, orders, and system configurations. Key features of the administrative interface include:
     - **Menu Management:** An intuitive interface for adding, editing, and removing menu items, including options for specifying item names, descriptions, prices, and availability status.
     - **Order Processing:** Tools for processing incoming orders, including order review, confirmation, and fulfillment functionalities, as well as options for managing order queues and assigning orders to specific staff members.
     - **User Management:** Administrative capabilities for managing user accounts, including user registration, account verification, and account status management.
     - **Reporting and Analytics:** Reporting tools and analytics dashboards provide insights into order trends, inventory management, and user behavior, enabling data-driven decision-making and performance monitoring.

Tailor this section to describe the specific user interfaces and interaction points relevant to your mess preordering system within the FoodXpress project, ensuring clarity and coherence in describing how users and administrators interact with the system.
.>

3.1.2 Hardware Interfaces

<

Hardware Interfaces

1. **User Devices:**
   - The mess preordering system is accessible via standard computing devices with internet connectivity, including but not limited to:
     - Desktop computers
     - Laptop computers
     - Smartphones
     - Tablets
   - The system is designed to be responsive and compatible with various screen sizes and input methods, ensuring a consistent user experience across different devices.

2. **Point-of-Sale (POS) Systems:**
   - Integration with existing point-of-sale (POS) systems may be required for processing payments and synchronizing order data with other operational systems within institutional dining facilities.
   - The system should support standard POS hardware peripherals such as:
     - Cash registers
     - Payment terminals (e.g., credit card readers)
     - Receipt printers

3. **Kitchen Display Systems (KDS):**
   - In some cases, integration with kitchen display systems (KDS) may be necessary for transmitting order details directly to kitchen staff for preparation and fulfillment.
   - The system should support communication protocols commonly used by KDS devices, ensuring seamless transmission of order information.

4. **Internet Connectivity:**
   - Reliable internet connectivity is essential for accessing the mess preordering system from user devices and transmitting data between the system and external servers.
   - The system should operate efficiently over standard internet connections, with considerations for bandwidth requirements and network latency.

Tailor this section to describe any specific hardware interfaces and dependencies relevant to your mess preordering system within the FoodXpress project, ensuring clarity in outlining the hardware requirements for system operation and integration.
>

3.1.3 Software Interfaces

<.>

3.2 Functional Requirements

<

Functional Requirements

*1. **User Registration and Authentication:***

- The system shall provide user registration functionality, allowing users to create accounts with unique usernames and passwords.

- Users shall be able to log in to the system using their registered credentials, with support for password recovery and account management.

*2. **Menu Browsing and Selection:***

- Users shall be able to browse through the available menu items categorized by meal types (e.g., breakfast, lunch, dinner) and cuisine types.

- The system shall display detailed information for each menu item, including item names, descriptions, prices, and availability status.

*3. **Order Placement and Customization:***

- Users shall be able to add selected menu items to their shopping cart, specifying customization options such as portion sizes, dietary preferences, and additional ingredients.

- The system shall validate user inputs and provide feedback on item availability, order constraints, and pricing adjustments in real-time.

*4. **Order Management:***

- Users shall have access to a centralized dashboard where they can view and manage their current and past orders.

- The system shall allow users to modify or cancel pending orders within a designated timeframe before the order cutoff time.

*5. **Checkout and Payment Processing:***

- The system shall facilitate secure checkout and payment processing for user orders, supporting multiple payment methods such as credit/debit cards and institutional meal plans.

- Users shall receive electronic receipts and confirmation emails upon successful order placement and payment processing.

*6. **Order Notification and Confirmation:***

- Users shall receive timely notifications and confirmations regarding the status of their orders, including order confirmation, pickup/delivery details, and updates on any changes or delays.

- Notifications shall be sent via email and/or SMS based on user preferences and contact information provided during registration.

*7. **Accessibility and Usability:***

- The system shall be accessible to users with disabilities, complying with WCAG (Web Content Accessibility Guidelines) standards and providing alternative navigation options for keyboard and screen reader users.

- User interfaces shall be designed with usability best practices in mind, ensuring intuitive navigation, clear feedback, and responsive layouts across different devices and screen sizes.

Tailor this section to include any specific functional requirements relevant to your mess preordering system within the FoodXpress project, ensuring clarity and completeness in defining the expected system behaviors and capabilities.. >

3.2.1 Functional Requirement #1: The system shall …

3.2.2 Functional Requirement #2

Description:
The system shall provide functionality for users to create a new account by providing required information such as username, email address, and password. Upon successful registration, the system shall generate a unique user ID and store the user's information securely in the database.

Rationale:
User registration is necessary to enable personalized features such as order history tracking, account management, and targeted communication. By allowing users to register for an account, the system can provide a tailored and enhanced experience for individual users.

Acceptance Criteria:
1. Users can access the registration page from the system's homepage.
2. The registration form includes fields for username, email address, password, and any other required information.

3. Upon submission of the registration form, the system validates user inputs and checks for duplicate usernames or email addresses.
4. Users receive a confirmation email containing a verification link to activate their account.
5. Users can log in to the system using their registered credentials after successful account activation.

You can continue to define additional functional requirements using the same format, incrementing the requirement number (e.g., F2, F3, etc.) for each new requirement. This approach helps organize and document the specific capabilities and behaviors expected from the system, making it easier for stakeholders and development teams to understand and implement the requirements effectively.

3.3 Use Case Model

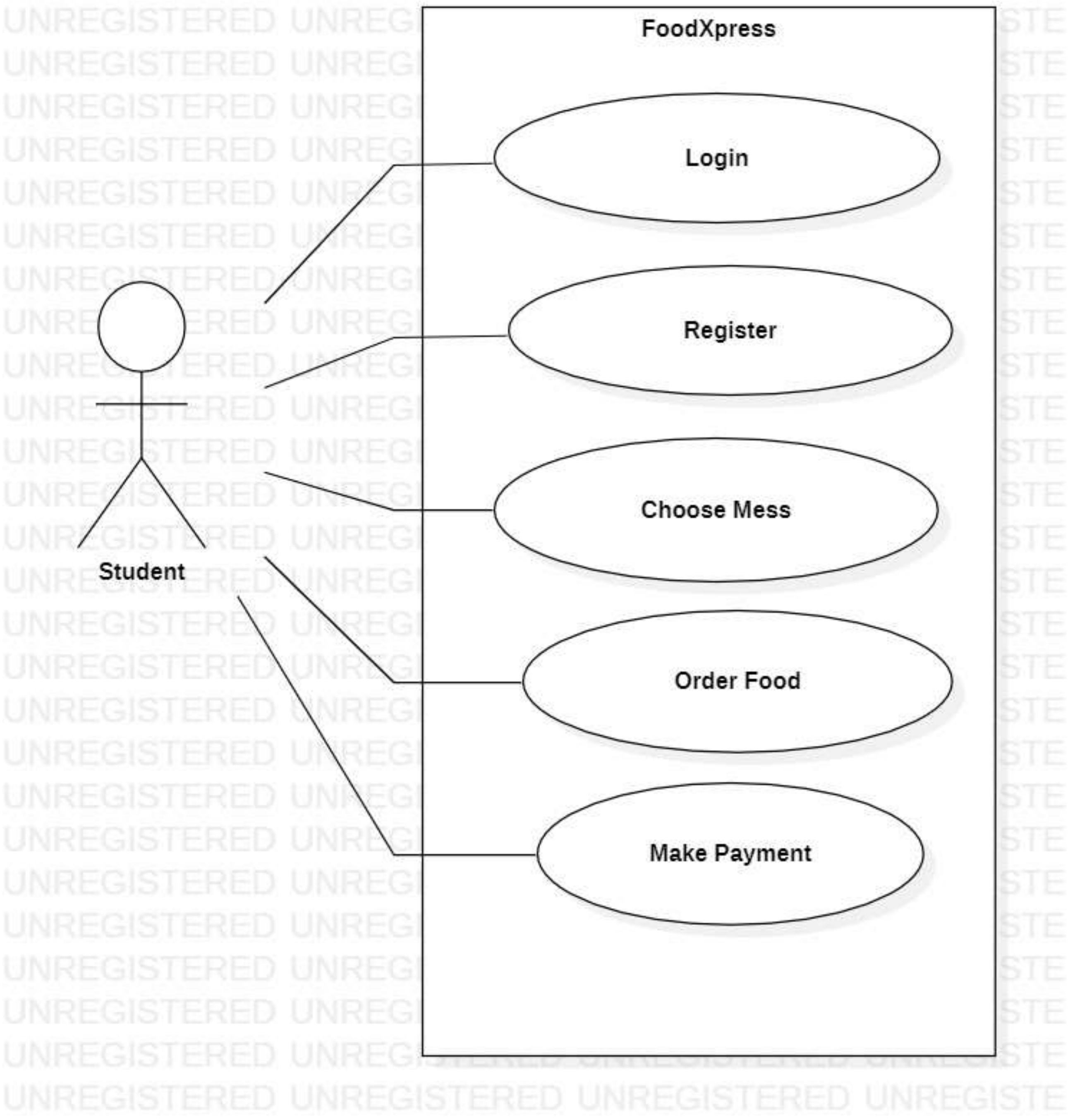

Figure 1: USE CASE DIAGRAM 1

3.3.1 Use Case #1 (Use Case Diagram 1)

Author – Purushottam Mahajan.

Purpose – Use case Diagram1 mentions the Student point of view of accessing the System. This Diagram outlines the activities processed by the Student as the user of the proposed system.

Requirements Traceability – All the requirements mentioned in the Use case are traceable and successfully measured by our team.

Priority -The priority of the USE CASE DIAGRAM 1 is defined as the High Priority. Importance of this use case being completed and functioning properly when system is deployed.

Preconditions – Any user using the system must be a registered user.

Post conditions – The user must receive a acknowledgement of confirmation of order from the system.

Actors –Students using the system are the actors mentioned in use case diagram 1.

Extends – NA.

Flow of Events

1. Basic Flow – Login or register to system, choose any mess and order food.
2. Alternative Flow - NA
3. Exceptions - NA

Includes (Use Case Diagram 2)

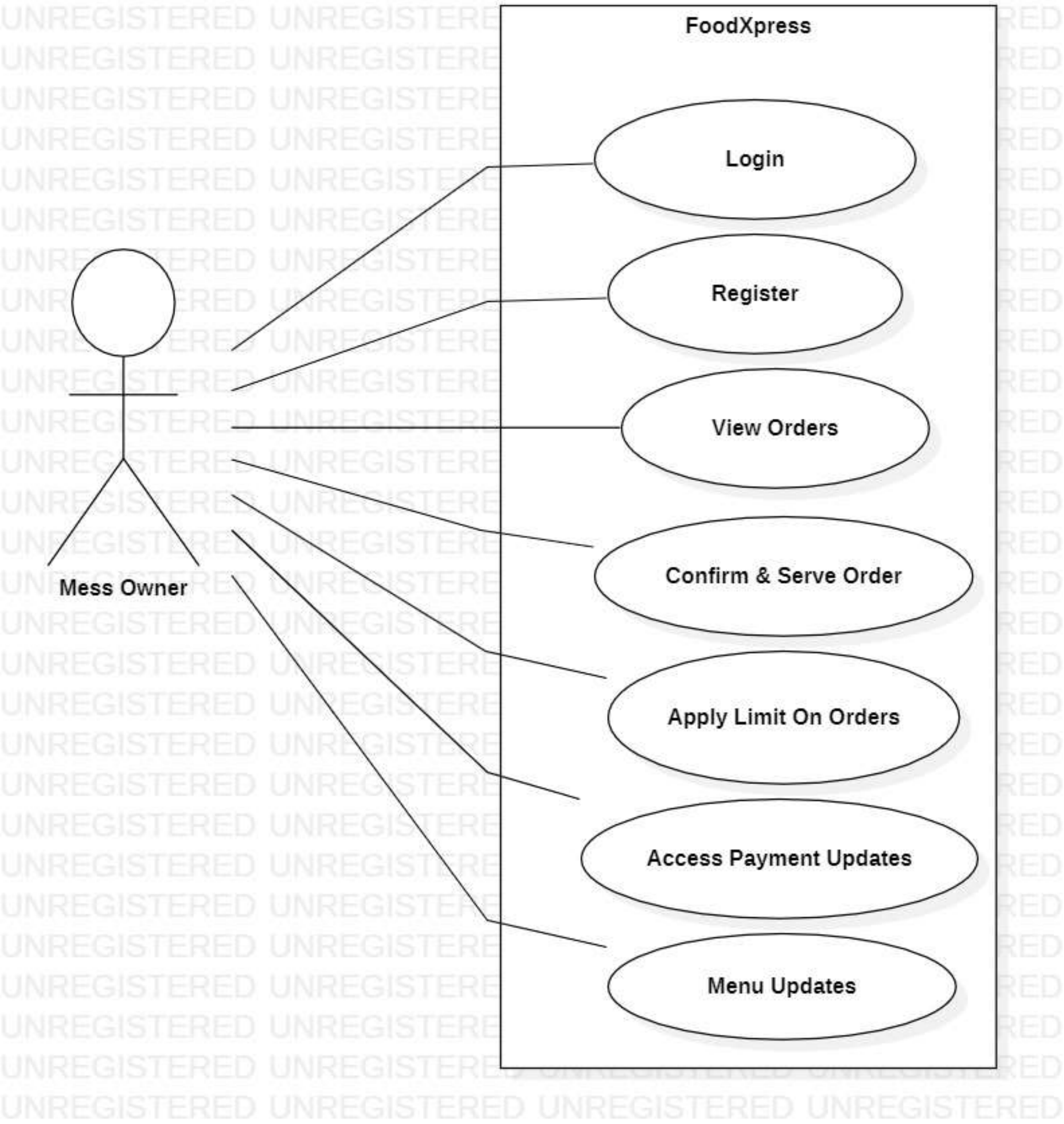

Figure 2: USE CASE DIAGRAM 2

3.3.2 Use Case #2(Use Case Diagram 2)

Author – Purushottam Mahajan.

Purpose – Use case Diagram2 mentions the Mess Owner point of view of accessing the System. This Diagram outlines the activities processed by the Mess Owner as the user of the proposed system.

Requirements Traceability – All the requirements mentioned in the Use case are traceable and successfully measured by our team.

Priority -The priority of the USE CASE DIAGRAM 2 is defined as the High Priority. Importance of this use case being completed and functioning properly when system is deployed.

Preconditions – Any user using the system must be a registered user.

Post conditions – The user must receive orders notification and payments updates.

Actors –Mess Owners using the system are the actors mentioned in use case diagram 1.

Extends – NA.

Flow of Events

1. Basic Flow – Login or register to system, Check orders and serve to students.
2. Alternative Flow - NA
3. Exceptions - NA

Includes (Use Case Diagram 1)

4 Other Non-functional Requirements

4.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.
TODO: Provide performance requirements based on the information you collected from the client/professor. For example, you can say "P1. The secondary heater will be engaged if the desired temperature is not reached within 10 seconds">

4.2 Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.
TODO:

- *Provide safety/security requirements based on your interview with the client - again you may need to be somewhat creative here. At the least, you should have some security for the mobile connection.*

4.3 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Adaptability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write "This software shall be maintainable…" Indicate how you plan to achieve it, & etc…Do not forget to include such attributes as the design for change (e.g. adapting for different sensors and heating/AC units, etc.). Please note that you need to include ***at least*** *2 quality attributes. You can Google for examples that may pertain to your system.>*

5 Other Requirements

<This section is ***Optional***. *Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.*>

Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>