খুলনা প্রকৌশল ও প্রযুক্তি বিশ্ববিদ্যালয়

প্রভু!
আমায়
জ্ঞান দাও

Course Title : Information Systems Design Laboratory

Course No : CSE 3120

Visit Report : Visited to Industry – Helium Restaurant, Khulna

Submit Date : 24-09-2025

Submitted To, Md Nazirulhasan Shawon Lecturer, Department of Computer Science and Engineering Safin Ahmmed Lecturer, Department of Computer Science and Engineering	Submitted By , Name : Ankon Roy Roll Group : 2107113 Year : B2 Semester : 3rd Department : 1st Visit Group : Computer Science and Engineering 1. 2107103 2. 2107105 3. 2107111 4. 2107113 5. 2107116

1. Objectives:

- To gain real experience of using the real-world System Development Life Cycle (SDLC) followed in software companies, and make a solution for the visited restaurant.
- To understand how we can use Agile methodology and Iterative Waterfall models to create real-world solutions.
- To analyze the system and create design diagrams such as DFDs, UML diagrams, ER diagrams, and Schema diagrams for the solution.
- To analyze how the restaurant interacts with customers, collect orders, collect raw materials, and finally serve them to the customers.
- To study challenges faced by the industry and learn what qualities are essential for future software engineers.

2. Introduction:

SDLC stands for Software Development Life Cycle. It is a structured process that defines the phases involved in planning, designing, building, testing, deploying, and maintaining high-quality software. The SDLC provides a systematic framework to ensure software is developed efficiently, meets user requirements, and is delivered on time and within budget. In our academic lessons, we study various models, including Waterfall, Agile, Iterative, and Spiral. But in industry practice, we often have to blend methodologies depending on the nature of the client requirements, project, and team expertise.

As part of our **CSE 3120 (Information Systems Design Laboratory)** course, we visited Helium, a restaurant and food serving company based in Khulna. The company serves food to customers at the restaurant and also has an option for online ordering. The purpose of our visit was to observe their work environment, analyze their business workflows, understand how they collect raw materials, prepare food, serve their customers, keep a log of their finances, track their employees' attendance and salary, and make a design with various diagrams using the knowledge I gained from the course.

3. Methodologies:

✞ SDLC Lifecycle for developing a new software for Helium Restaurant:

From our visit, we observed that the company needs a management software. Based on the project requirements, we can use these two methods:

- **Agile Methodology:** For dynamic projects with continuous client feedback. Iterations are short (sprints), and customer feedback is integrated regularly.

Steps for developing our management software:

1. **Requirement Analysis**
   - Initial meetings to gather requirements with the restaurant owner and managers.

- Sometimes clients lack technical expertise, so we will prepare mockups or prototype demos.

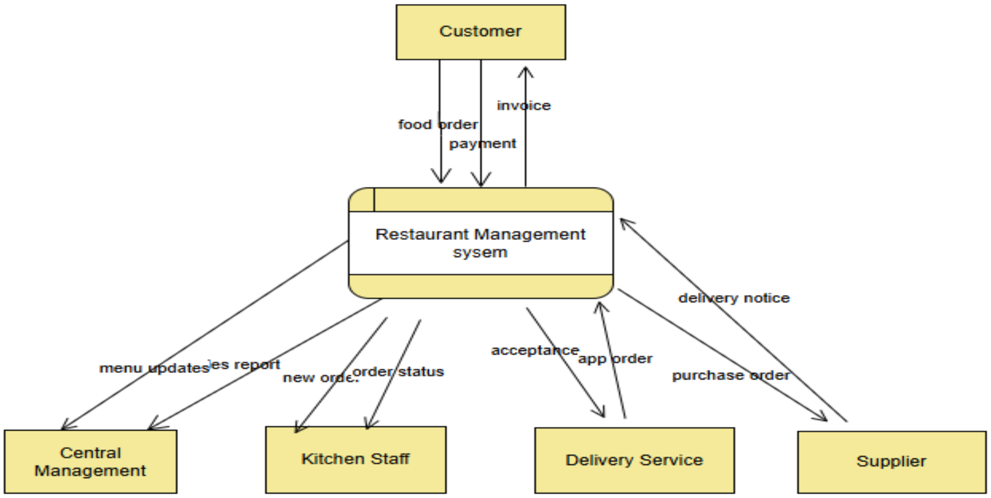

Fig-1: DFD Diagram (Context Level)

2. **System Design**
    - We will prepare mockup UIs and system architecture diagrams.
    - We will use DFDs to model system processes.
    - Use UML Use Case Diagrams to show interactions between actors and the system.
    - ER Diagrams and Schema diagrams for database design.
3. **Implementation / Development**
    - We will divide the Work among developers (team-based).
    - Collaborate using version control (Git/GitHub).
    - Manage the coordination of Frontend and backend teams using APIs.
4. **Testing**
    - Testing is a key step before deployment. We will perform unit, integration, and user acceptance testing (UAT), along with bug tracking, to ensure system reliability. In Agile, testing is continuous within each sprint, while in the Iterative Waterfall, it is done after full implementation.
5. **Deployment**
    - We will deliver the project either after multiple agile sprints (for Agile) or in one package (for Iterative Waterfall).
6. **Maintenance**
    - If minor changes are required post-deployment, the team makes fixes.
    - Continuous updates provided for agile projects.
    - New features will be added based on new requirements in the future.

✞ Diagram Usage in Practice:

- **DFD (Data Flow Diagram):** Used for requirement analysis and modeling the flow of data between customer, system, and databases.

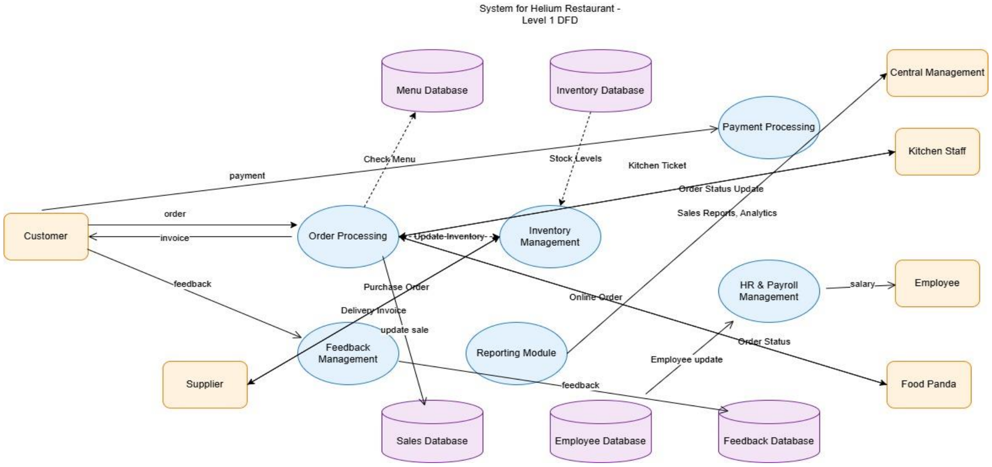

Fig-2: DFD Diagram (Level-1)

- **Use Case Diagram:** Used to show functional requirements from the user's perspective.

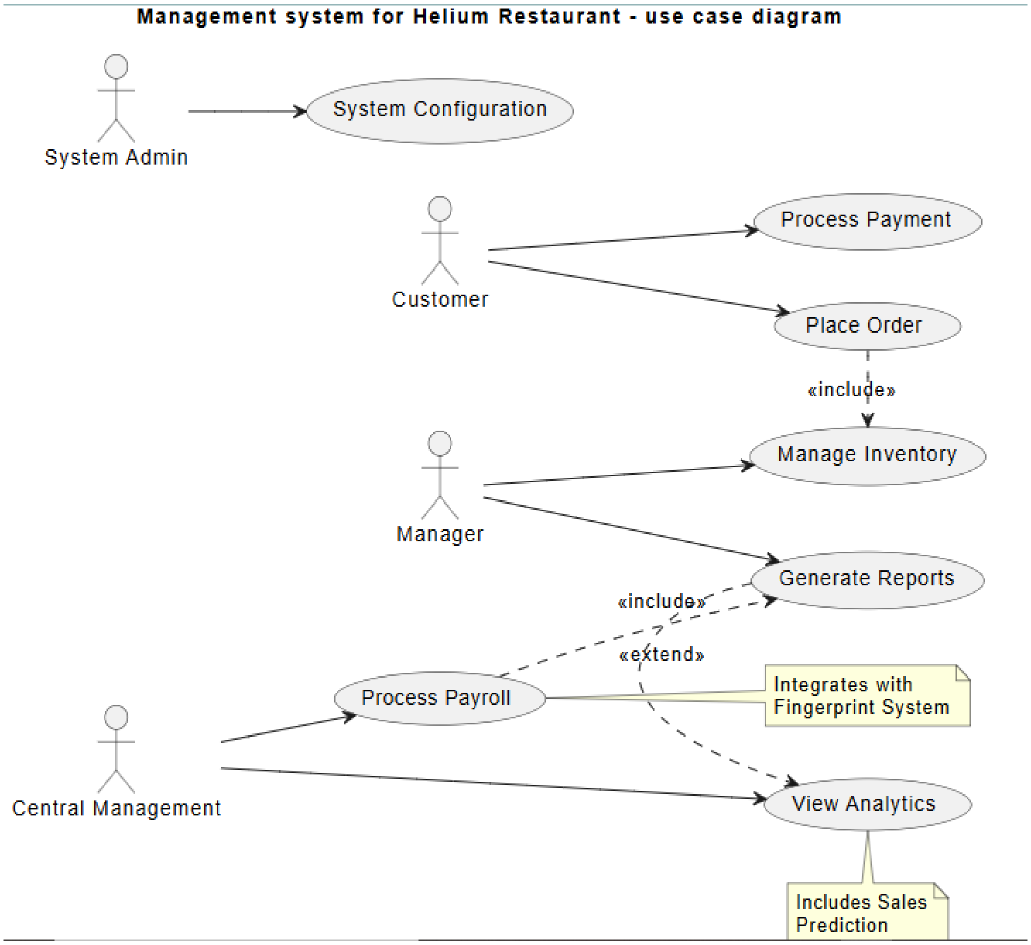

Fig-3: Use Case Diagram

- **Class Diagram:** Shows the classes and their relations used in the management system.

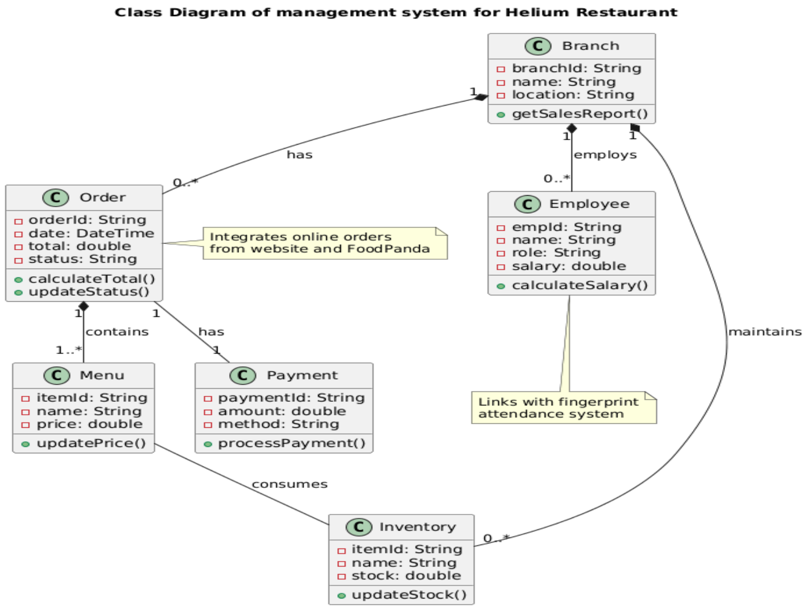

Fig-4: Class Diagram

- **Sequence Diagram:** Sequence diagram for our restaurant management system.

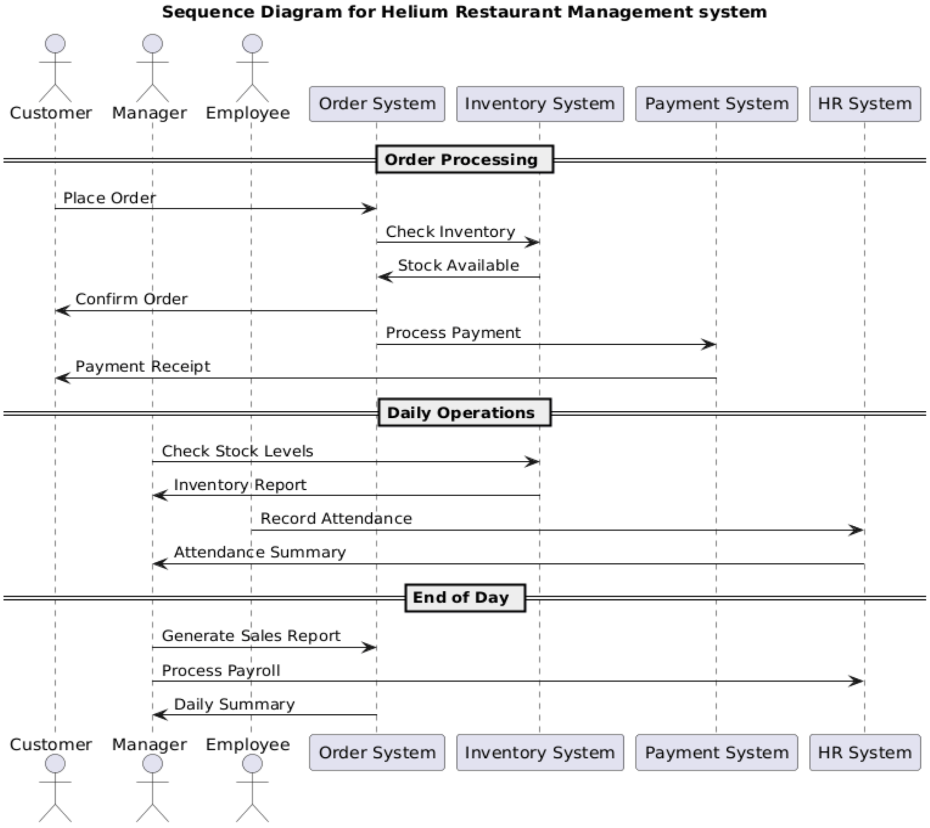

Fig-5: Sequence Diagram

- **Activity Diagram:** Activity diagram for our restaurant management system.

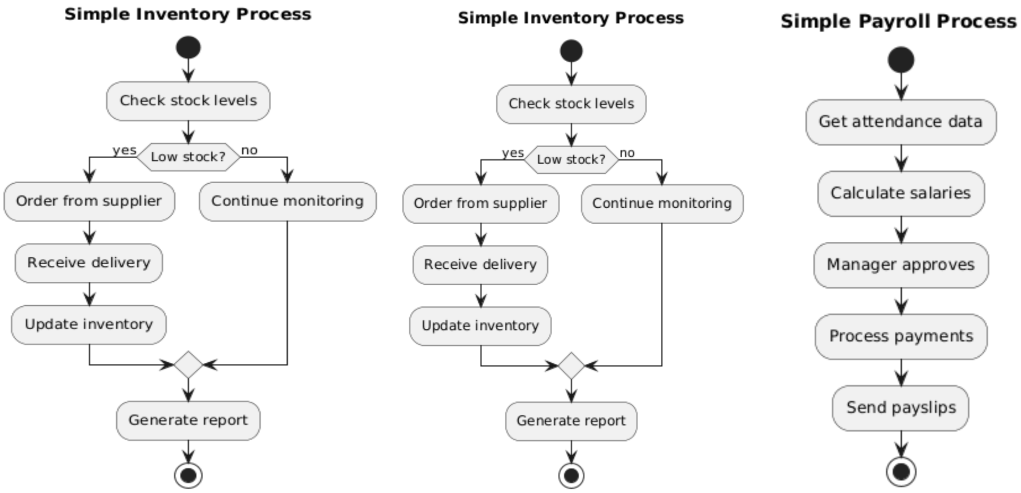

Fig-5: Activity Diagram

4. Testing:

We emphasize that testing is a critical step before deployment. Our testing strategies will include:

- **Unit Testing:** Individual modules will be tested by developers.
- **Integration Testing:** Interfaces between frontend, backend, and database will be tested.
- **User Acceptance Testing (UAT):** Users will interact with the system and provide final approval.
- **Bug Tracking:** Issues will be identified, logged, and resolved before deployment.
- In Agile projects, testing happens continuously within each sprint.

4. Discussion:

From our visit to Helium Restaurant, we observed a common challenge faced by growing businesses, and that is evolution from ad-hoc, manual processes to a streamlined, integrated system. The restaurant currently operates in a hybrid form of digital and analog operations, which, while functional, creates significant operational friction and data mismanagement. The core problem is both a lack of technology and a lack of central integration. They use computer technologies like 3S software for keeping log for food sales, the fingerprint scanner for employee attendance, and the online portals for displaying and taking online orders. But all of them operate in isolation. Also, they don't use any kind of automation tool for keeping the logs of their employees' salaries and the buying of their raw materials. This forces the staff to fill the gaps with their manual labor. This leads to inefficiencies, a high potential for error in inventory and payroll, and a critical inability to gain a consolidated view of business performance across all the branches of their company.

That's why we proposed a new system that is designed specifically to address this disintegration and lack of automation. By using standard software engineering methodologies like UML diagrams and DFDs, the proposal moves from a vague idea to a structured blueprint. The Use Case and Activity diagrams clarify how all users—from cashiers to central management—would interact with a unified system, eliminating redundant data entry. The Sequence Diagram highlights the seamless, automated flow of an online order, demonstrating a significant improvement in accuracy and speed over the current fragmented process.

Finally, one of the most significant benefits of this integration is the transformation of data into actionable intelligence. Currently, sales prediction is impossible because the data is scattered. But our new system would automatically process sales, inventory, and feedback, enabling data-driven decision-making. Management could identify popular menu items, forecast demand, optimize stock levels, and tailor marketing strategies based on centralized customer feedback. Also, by using various software development methods (Agile), we will be able to implement new features and keep our business growing and up to date with current world

5. Conclusion:

In conclusion, this project underscores a fundamental principle in business technology, computer system automation, and a central computer system integration is a must in today's world. For Helium Restaurant, implementing an integrated and automated system is not merely an IT upgrade; it is a strategic move towards operational excellence, informed management, and scalable growth. It would replace manual effort with automated workflow, and guesswork with insightful analytics, ultimately creating a more resilient and competitive business.

The visit reinforced the importance of:

- Agile practices in handling uncertain and evolving requirements.
- DFDs and UML diagrams in clarifying communication between developers and customers.
- Testing and maintenance as critical phases of SDLC.

Overall, the experience helped us to use the knowledge that we gained in our classroom to use in real-life scenarios. While working to solve a real-life problem, we identified and filled the gap between our classroom learning and industrial practices, providing us with better insight into the skills and practices required to excel as future software engineers.