**Week4 Agile and DFD Exercises 1 & 2**

Hang Yang

SYS 5013: System Engineering Analysis

Prof. Hitam Zidan

Sep 13, 2024

Contents

[Exercise1: Food2Go Delivery Service Prototype Report 2](#_Toc177311363)

[Excersie2: DFD on Student Registration System Report 6](#_Toc177311364)

# Exercise1: Food2Go Delivery Service Prototype Report

This project involves building a mobile food delivery service prototype using Agile modeling techniques and other essential tools. Food2Go, an expanding food delivery company, has a requirement to redesign and upgrade its current system to gain an advantage in the highly competitive market. To achieve this, we will employ key information-gathering methods-interviews/surveys, use case diagrams, Agile modeling, and prototyping to design a next-generation delivery system. In this report, we will focus primarily on the Agile modeling process. Therefore, details related to interviews, surveys, and use case scenarios will not be included. Instead, we will use the use case diagram as a starting point to directly develop our user stories, which will guide the rest of the system development process.

**Prototype Canva Link**: <https://www.canva.com/design/DAGQwG669yM/LejXuzyXJk0C8ifrZlh85w/edit?utm_content=DAGQwG669yM&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

**Agile Modeling Techniques**

**Meet with Users to Collect Requirement Information**

As mentioned above, the detailed information-gathering process will not be covered in this report. However, we would employ a combination of interviews, surveys, observations, and historical data analytics to build a comprehensive understanding of the business’s pain points and identify new revenue opportunities in the redesign. This multi-faceted approach allows us to gain insights from both stakeholders and data, ensuring that the new system addresses existing challenges while opening new possibilities for growth and efficiency.

**Logical Use Case Diagram**

Following the information-gathering process, we will begin by constructing a logical use case diagram to visualize the interactions between key users of the system. This diagram will facilitate understanding of how different user roles—such as food consumers (customers), food providers (restaurant managers), food deliverers (drivers), and the system scheduler—interact with the system. The system scheduler plays a crucial role in coordinating the dispatch of food orders from restaurants to available drivers, ensuring timely and efficient deliveries. This use case diagram will serve as a foundation for developing user stories and ensuring that each actor's needs and interactions are clearly defined in the system design.

A diagram of a network

Description automatically generated with medium confidence

*Figure1.* Food Delivery Service Use Case Diagram (Source: own photo)

**Documented Covered Features using User Stories**

***User Authentication***

* As a customer, I can log into the food delivery system so that I can place orders and track deliveries.
* As a customer, I can recover my password so that I can regain access to my account.

***Home Page and Search***

* As a customer, I can view promotions and my frequent shopping meals on the homepage so that I can quickly choose from the most preferred options.
* As a customer, I can search for restaurants by name, food type, or address so that I can find a restaurant that fits my preferences.
* As a customer, I can browse the searched restaurants which are open at the time and ordered by their rating scores so that I can make an informed decision.
* As a customer, I can see the real-time order tracking progress so that

***Restaurant and Menu Interaction***

* As a customer, I can select my desired restaurant so that I can view its menu.
* As a customer, I can view the entire menu of a selected restaurant so that I can place meal orders.
* As a customer, I can add meals to my shopping cart so that I can prepare my order before checkout.
* As a customer, I can review my shopping cart, including meals, subtotals, tax, and totals, so that I can verify everything before payment.
* As a customer, I can edit meals in the shopping cart by adding more meals or deleting mistakenly selected meals so that I can finalize the correct order.

***Order Placement and Shipping Information***

* As a customer, I can add my shipping details (e.g., name, address, phone number) and save it for future use so that I don’t have to enter it repeatedly for each order.
* As a customer, I can specify my preferred delivery time so that I can coordinate the arrival of the order with my schedule.
* As a customer, I can enter and save payment information so that I can quickly check out future orders.
* As a customer, I can receive a digital receipt with the order confirmation number after my payment is successful so that I have a record of my transaction.

***Order Tracking and Delivery***

* As a customer, I can track the delivery status, including estimated waiting time, so that I know when to expect my food.
* As a customer, I can receive notifications when the food is delivered so that I can pick it up promptly.
* As a customer, I can leave feedback on the delivery service and food quality so that I can share my experience and help improve the service.

***Restaurants, Drivers, and System Schedulers***

* As a restaurant manager, I can prepare meals based on the order placed so that the food is ready for delivery.
* As a system scheduler, I can assign a driver to the order so that it can be delivered to the customer.
* As a driver, I can pick up food from the restaurant so that I can deliver it to the customer.
* As a driver, I can deliver the food to the customer’s address so that the order is completed.

**Summary**

To build a food delivery service platform, I would focus on Agile modeling techniques to ensure flexibility and continuous user feedback. The process would start with user interviews to gather detailed user stories from customers and stakeholders. Given the highly interactive nature of the platform, direct interviews would provide valuable insights into user needs, helping refine early designs and align the product with customer expectations. Next, I would create a use case diagram to visually map user interactions with the system. This diagram would serve as a high-level guide, ensuring all key functionalities and scenarios are captured early in the development process. From this, I would write detailed user stories and organize them into a Product Backlog, forming a clear roadmap for development. For prototyping, I would use Figma, a no-code platform that allows for collaborative design. Figma helps create early-stage display prototypes, giving stakeholders a visual representation of the system’s interface. This early feedback loop would ensure the system evolves in line with user expectations. Optionally, I could form a Scrum team, using Jira to manage tasks and sprints efficiently and DevOps practices for easy scalability and minimizing hardware maintenance costs. This overall approach emphasizes Agile's iterative development, ensuring the product remains flexible and responsive to user feedback throughout its lifecycle.

# Excersie2: DFD on Student Registration System Report

This report outlines the Physical Data Flow Diagrams (DFDs) for the refined design of the University Student Registration System. The existing legacy registration system suffers from common issues such as outdated functionality, messy user interfaces, and inconvenient processes. Our proposed redesign offers a potential solution to these challenges. The system design focuses on key functions such as course listing display, student enrollment, billing, and related data storage.

By analyzing the system from the top context diagram down to the detailed decomposition level, we have established clear and well-defined boundaries around the system’s four main partitions. These partitions effectively group related processes into manageable modules, reducing system complexity, enhancing maintainability, and improving overall performance. Partitioning ensures that each module functions independently, which allows for smoother updates, better scalability, and easier troubleshooting.

**DFD Canva Link**: <https://www.canva.com/design/DAGQxys1R1Y/C105oEP_KY2v9tn4fuef9A/view?utm_content=DAGQxys1R1Y&utm_campaign=designshare&utm_medium=link&utm_source=editor>

**Authentication Process**

The first partition is the Authentication Process, which plays a critical role in validating the user’s identity and authorizing their actions within the system. This partition ensures that students can access their personalized dashboard to view registered courses, billing information, and available courses, but limits their access to other functions, such as editing course details or viewing teaching staff schedules. Partitioning authentication process ensures security by isolating access control from other modules, protecting sensitive data while enhancing the user experience by guiding students directly to permitted actions.

**Student Dashboard Process**

The second partition is the Student Dashboard Process, where students can quickly access frequently used features. This includes viewing registered courses, checking remaining required credits, and performing shortcut operations such as paying bills or accessing course schedules. Partitioning this dashboard allows for modularity in the design, making it easier to update frequently accessed information without disrupting other system components. This process also ensures quick access to real-time data, improving the overall user experience by providing relevant information briefly.

**Course Selection and Enrollment Process**

The third partition, which is the most critical, is the Course Selection and Enrollment Process. This process is often a pain point in legacy registration systems, where students are left confused due to outdated course availability information. Partitioning this process ensures that course selection is treated as a prerequisite for enrollment, enforcing logical steps in the workflow. The system provides real-time feedback on course availability, eliminating confusion and improving student satisfaction. Additionally, we have enhanced the process by introducing filters and fuzzy search options for course number, course name, and status, allowing students to easily find desired courses and make informed decisions. This partition also ensures that updates in course availability are processed instantly.

**Billing Process**

The fourth partition is the Billing Process, which allows students to view their tuition balance, download bill statements in PDF format, and make payments. This process is isolated to handle specific financial transactions, ensuring security and proper error handling. For example, if a student's credit card is invalid, expired, or has insufficient funds, the system will reject the transaction and notify the student with clear guidance on how to resolve the issue. Additionally, students will receive payment confirmations once a valid transaction is completed. By partitioning the billing process, the system ensures that financial operations are separate from other student activities, thereby improving reliability and minimizing the risk of errors or fraud.

**Summary**

This Physical DFD of student registration system breaks down key business activities, such as login, course selection, and enrollment, into smaller, manageable sub-processes. This approach ensures a deep understanding and effective handling of complex tasks like course capacity management and validation. Integration with Master Data Stores is central to the design, ensuring seamless interaction with the Student Master, Course Master, and Degree Program Master. Real-time data retrieval and updates enable accurate processes, such as fetching course listings and validating credit limits. Error Handling and Validation steps are built into each process, ensuring robust system performance even in edge cases. Password verification, course availability checks, and concurrent enrollment handling all contribute to a smooth user experience. Finally, the design emphasizes scalability and modularity, allowing new features to be easily added or modified without disrupting the entire system. This ensures flexibility for future growth and adaptability to changing requirements.

**References**

Doodle Legend. (2023, June 18). How To Use Canva Prototyping Tool - Easy to Use Tool [Video]. YouTube. <https://www.youtube.com/watch?v=D-UtpQ8jVRg>

GeeksforGeeks. (2024, May 14). Types and Components of Data Flow Diagram (DFD). *GeeksforGeeks*. <https://www.youtube.com/watch?v=D-UtpQ8jVRg>