

Term Project: Dewhurst Dining Hall Mobile Application

Murielle Chaghouri & Mohamad Nour

Williams School of Business, Bishop's University
Information Technology Project Management Course

December 15th, 2024

System Request

Sponsor:

The sponsor of the project will be Bishop's University Dining Services, specifically the Dewhurst Dining Hall. The main goal is to improve the dining experience for students while improving operational efficiency by introducing a new mobile application.

Business Challenges:

Dewhurst Dining Hall currently faces a lot of difficulties during peak dining hours. For example, students usually order directly at food stations, and that make students wait for a long time, so there is overcrowding because of the high demand. Students and staff are not happy with this process. Because there is no system for this process, there is a lot of delays which reduces the quality of the dining experience.

Required Tasks to Resolve the Problems:

We will need to do the following tasks to solve these problems:

- We will need to develop a mobile application that allows students to browse menus, select items, and place orders for their meals.
- We should include an effective order management system for staff to track and prepare meals on time.
- In addition to that, we need to provide real time notifications to students for the status of their order pickups.
- Furthermore, staff must be trained well for this system.

Benefits of the New System:

- Students will not have to stand in long lines which will save time and improve their overall experience.
- The dining hall staff can prepare meals more effectively with a streamlined order management system.
- Both students and staff will benefit from a more organized and less stressful environment, so they will have higher satisfaction.
- The system will make peak hours easier to manage by spreading orders across time slots.
- The suppliers can track inventory to make sure that the hall is always well stocked.

Possible Constraints:

- The system might need technical maintenance like fixing bugs and updates, which could increase operational costs.
- The budget may impact the development and implementation process.
- The system needs to be scalable to handle heavy traffic load during peak dining hours with no crashes or delays.

Planning Phase

Technical Feasibility:

- Platform: App needs to be accessible on mobile devices (iOS and Android) by either developing a separate app OR choose a cross-platform framework.
- Backend and Database: Real-time backend system that tracks food availability, order, and notifications. Database also needed to store dynamic data (menus, orders, etc.).
- Real-time Tracking: Update students with order status, manage approximate wait times at food stations, and send notifications when food is ready to be picked up.
- Security: Handling students' sensitive information

Economic Feasibility

- Development cost: Hiring skilled staff to develop mobile app (20,000-30,000\$)
- Maintenance cost: Server costs to host the app and database, fix bugs, updates and optimize (2,000-5,000\$/year)
- Operational savings: Help reduce food waste by limiting number of orders at a time
- Funding: Student contribution with the meal plan fee they already pay, university IT budget, etc.

Organizational Feasibility

- Stakeholder support: Students need to use app regularly for it to be a success. The cafeteria staff also needs to be trained to manage incoming orders, monitor food availability, and update menus.
- Training: Staff will need to learn and train on how to deal with incoming orders which might take some time.
- System Integration: App should integrate with existing systems, need collaboration with IT, and staff must be open to change.
- Implementation Timeline: Few months for full implementation and time it during less busy periods (i.e.: during breaks) to minimize disruptions.

Work Plan:

- Gathering requirements: identify current pain points that app should address
- Design and prototyping: create app design and develop prototype for initial feedback
- Development: app core functionalities, development, and database structure
- Testing: test for bugs, performance, and user testing
- Launch and training: launch to app store and provide training to cafeteria staff
- Post-launch support: ongoing checks and maintenance to keep app up to date

Time Estimation:

Going through all the stages of the work plan, the project should take around 3 to 4 months total.

Staffing:

Project manager – Oversees the project and coordinates communication between staff

Developers – People familiar with required platforms and knowledge in backend

UI/UX designer – Someone to create interface easy to use for student and staff

Quality assurance tester – Someone testing the application and detect potential bugs or performance issues

Cafeteria staff representatives – People offering feedback and input to check system works properly daily

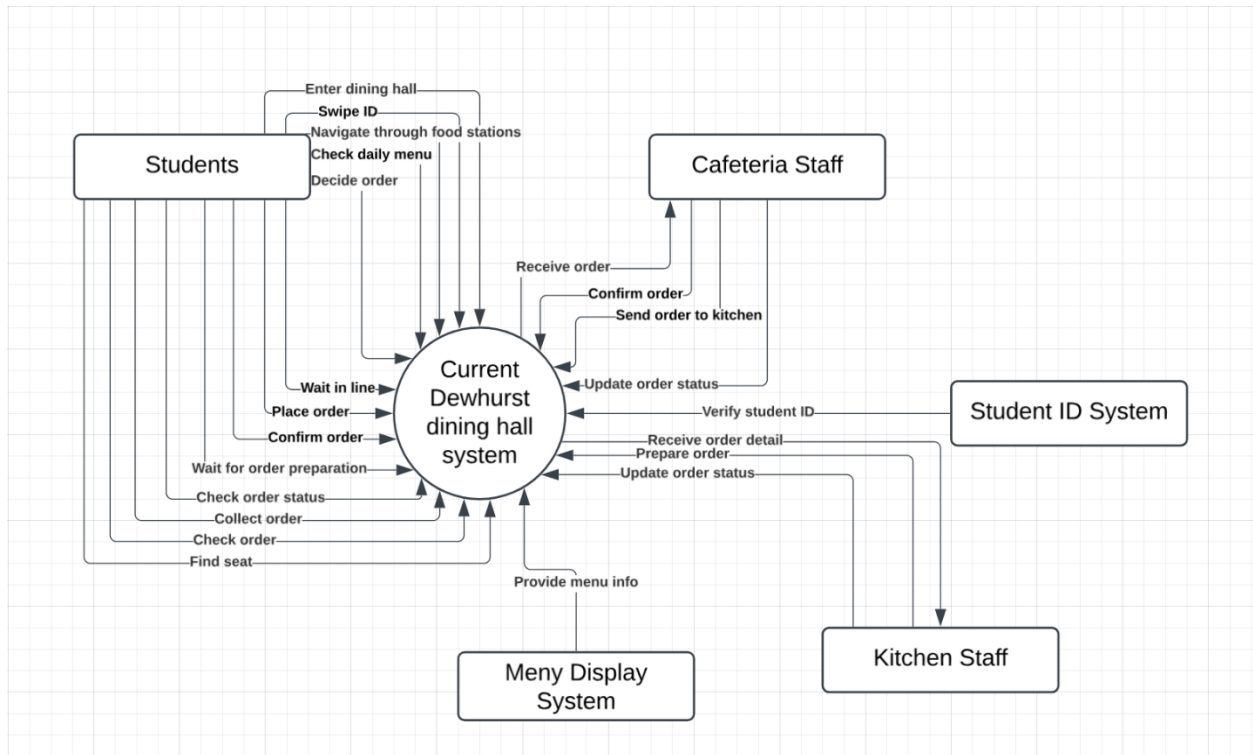
Techniques Used to Determine System Requirements

To determine system requirements, we used various techniques:

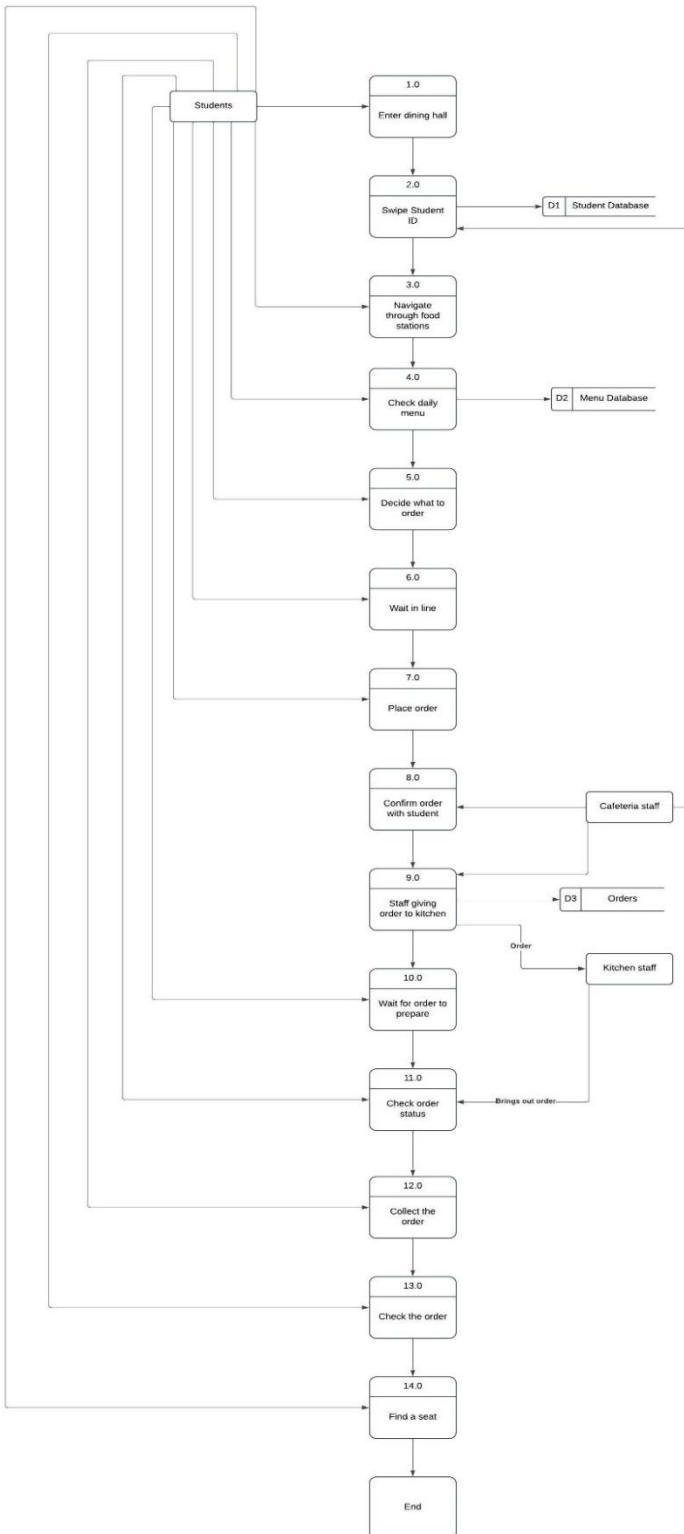
- Observations: While being frequent users of the dining hall, through observation we noticed inefficiencies in the current system they use, leading to high traffic in the dining hall, especially when ordering from food stations (grill, stir-fry, etc.) and problems with food waste management (i.e.: users not picking up their order). Therefore, our system should have a functionality to limit amount of order at a time per student.
- Interviews: By talking to daily users of the dining hall, they reported that navigating in the dining hall, checking menus, waiting in lines, and placing orders creates high traffic and wastes time. Therefore, the system should be able to show the menu, provide estimated wait time, and real-time notifications when order is received/preparing/ready.
- Personal experience: As frequent users of the dining hall, we agree that the current system has a lot of flaws, mainly a substantial amount of wait time, and complex process to order (which should be a simple task). Therefore, our system should show what is available and with a few clicks, place an order.
- Discussions: Through brainstorming and discussions, collaboration is crucial to determine and refine system requirements. Getting feedback from student and cafeteria staff helps us get more ideas on our proposed system and make it as best as possible.

Using those techniques provides us an understanding of the system requirements. They offer insights, helps ensure requirements and lets us address the expectations of students and cafeteria staff.

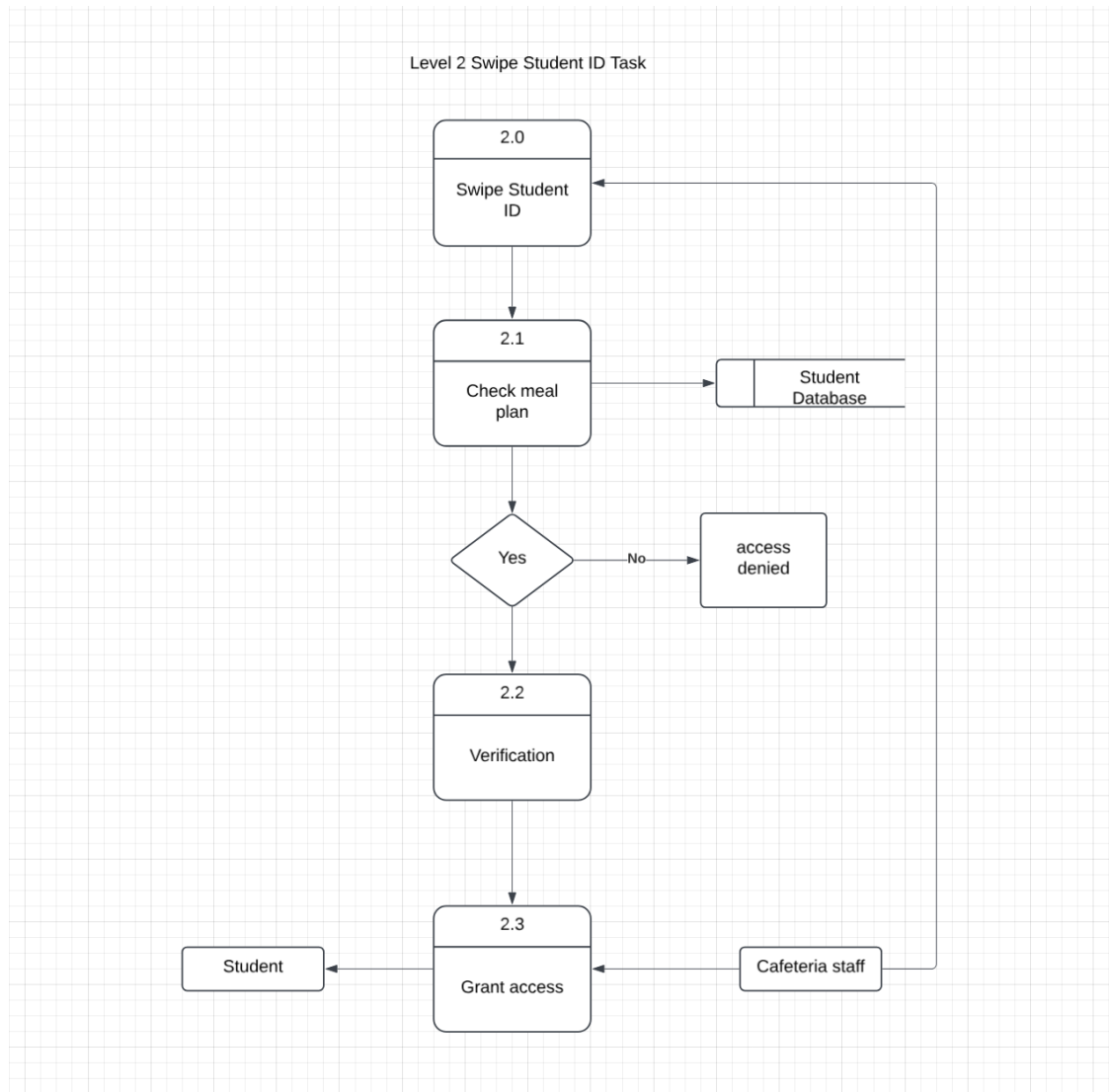
Context DFD For The AS-IS Business Process

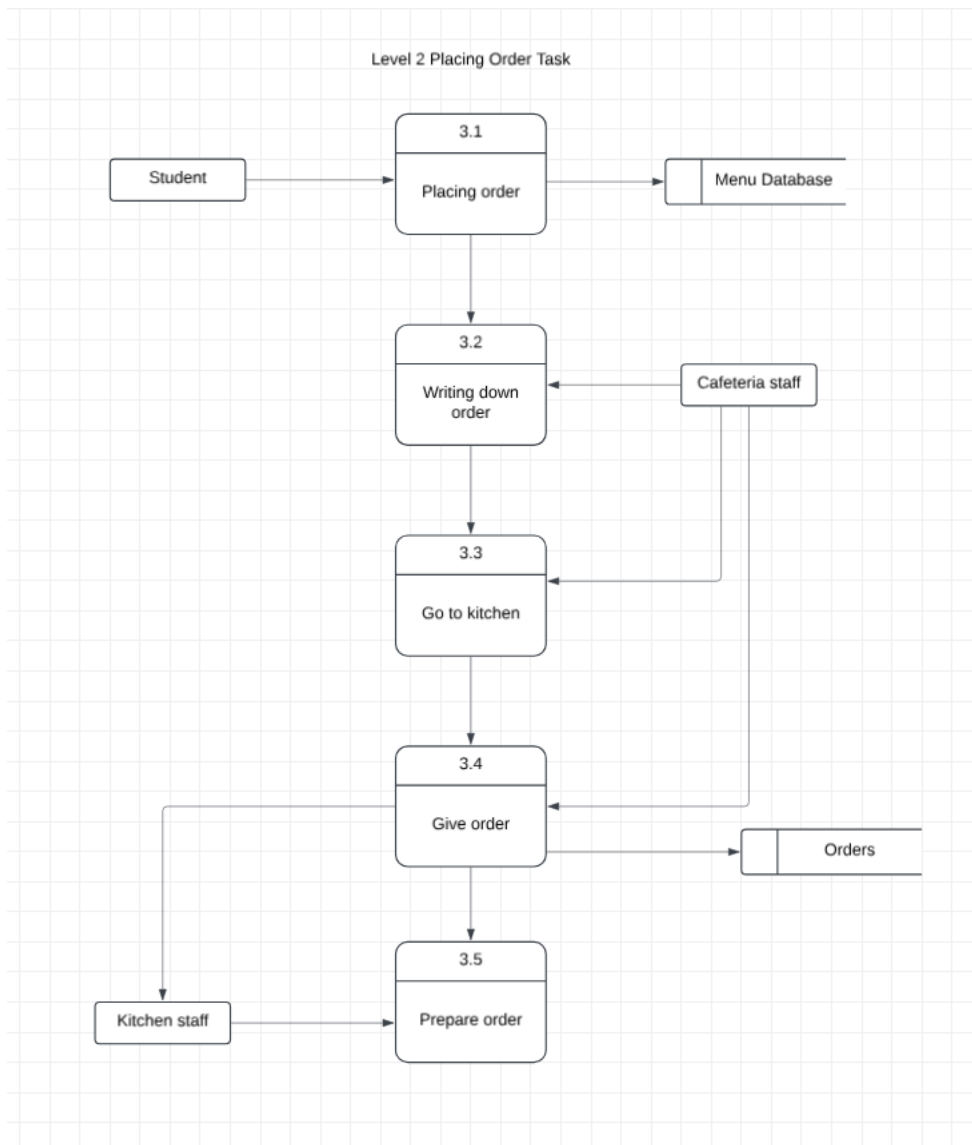


Level 1 DFD For The AS-IS Process

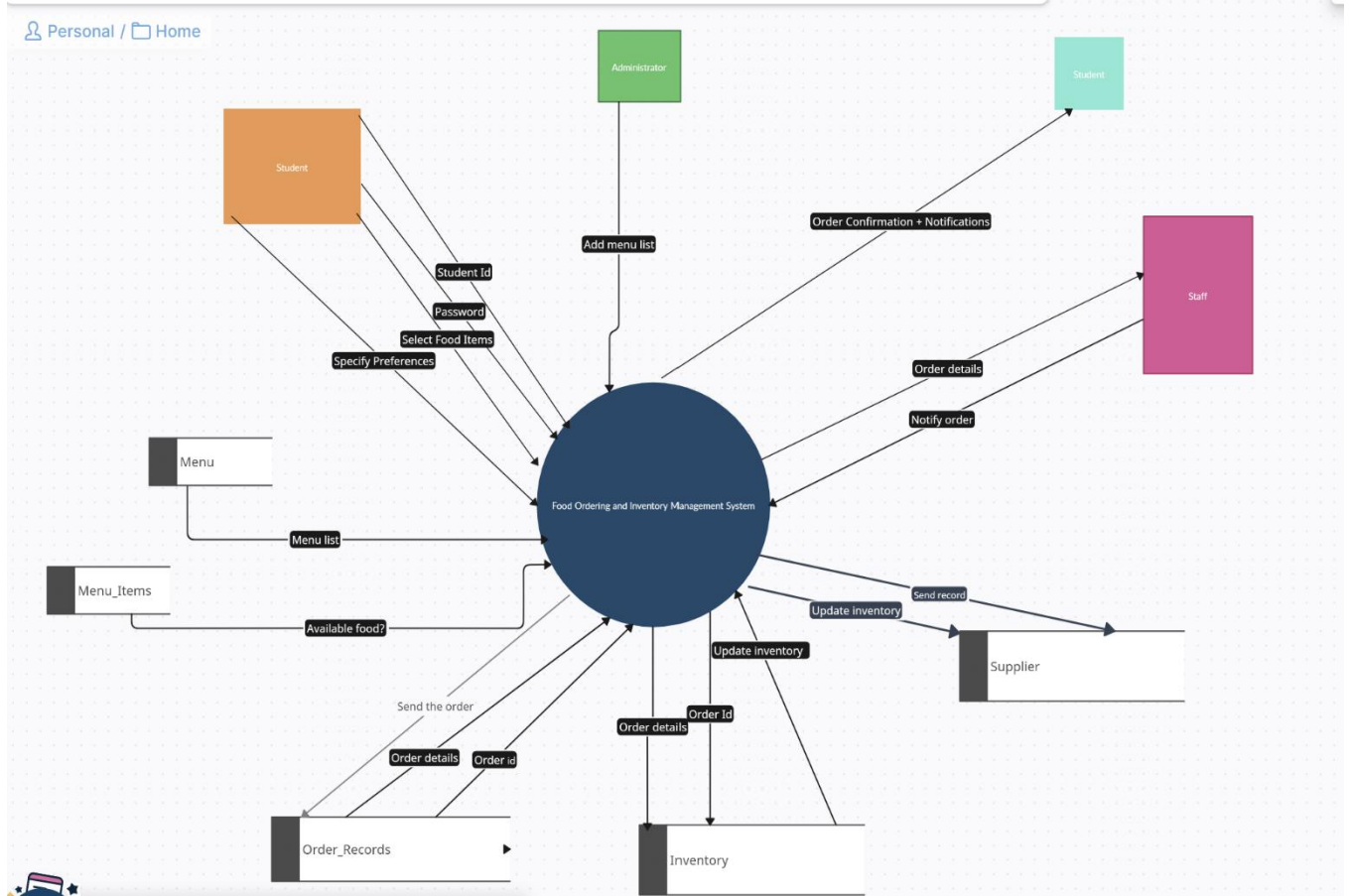


Level 2 DFD For The AS-IS Process (only for two tasks in level 1)

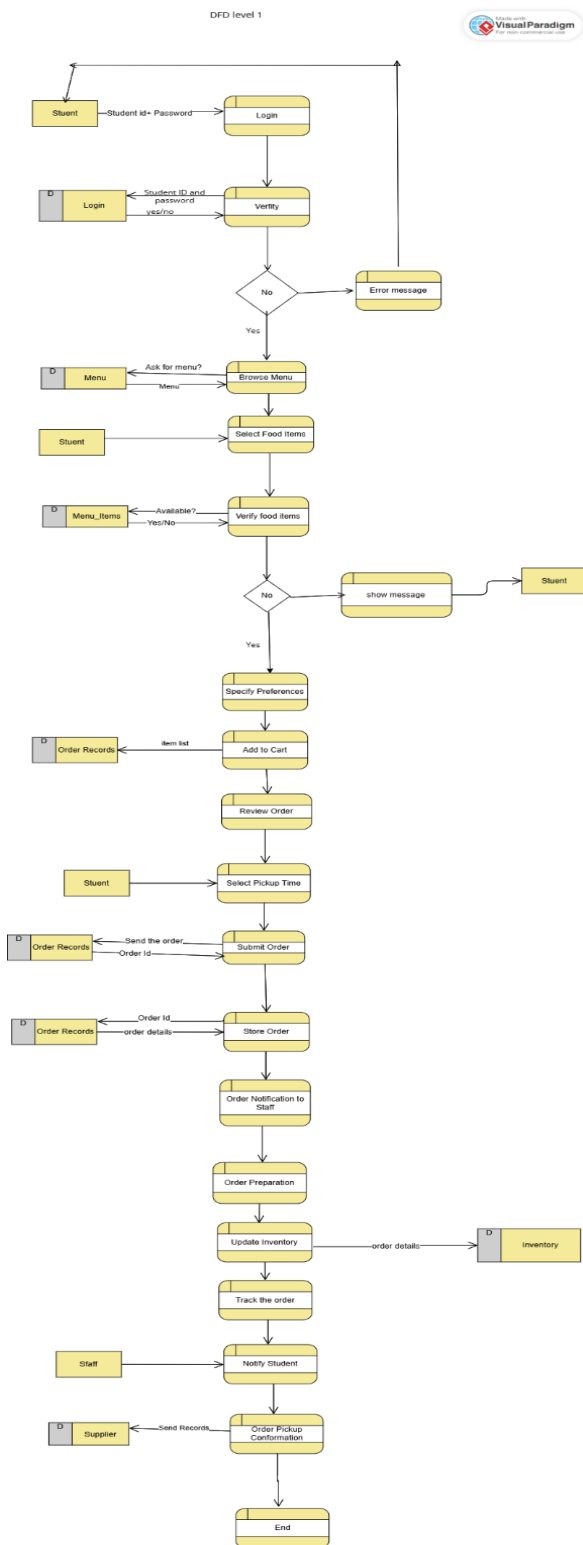




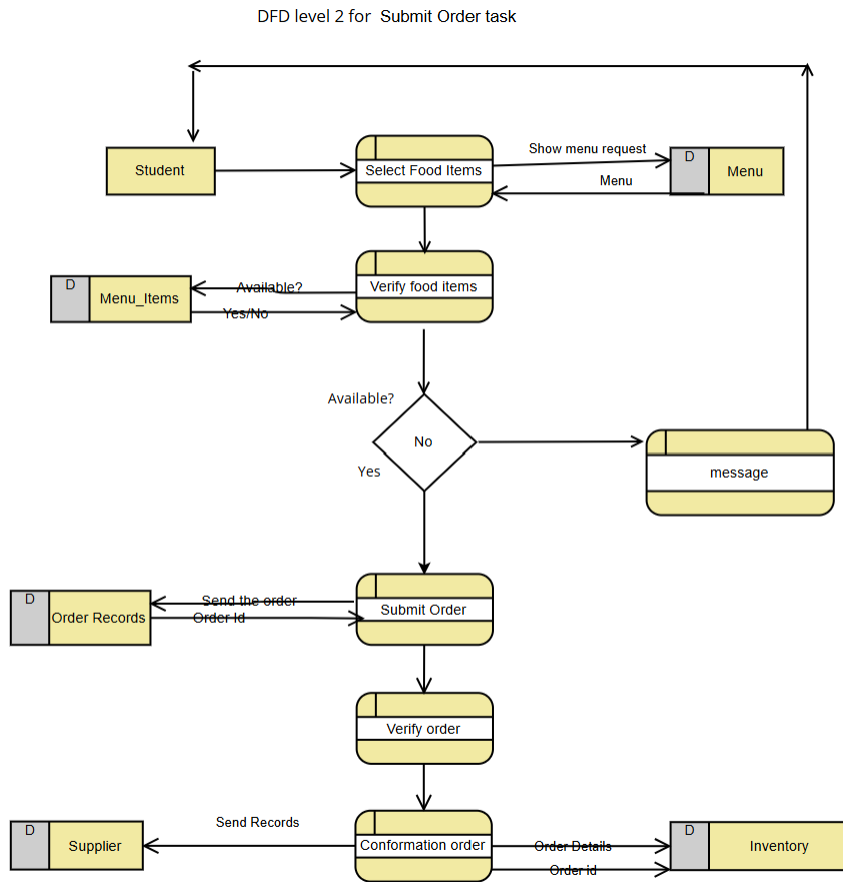
Context DFD For The TO-BE Process



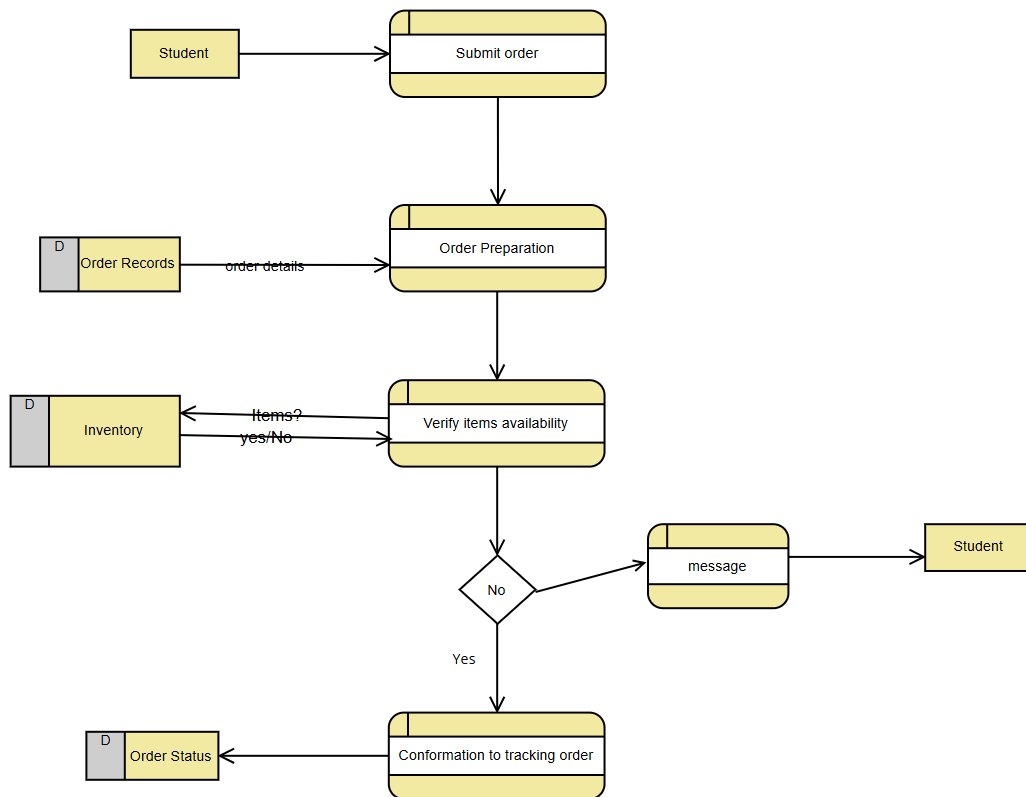
Level 1 DFD For The TO-BE Process



Level 2 DFD For The TO-BE Process (only for two tasks in level 1)



DFD level 2 for Order Preparation task

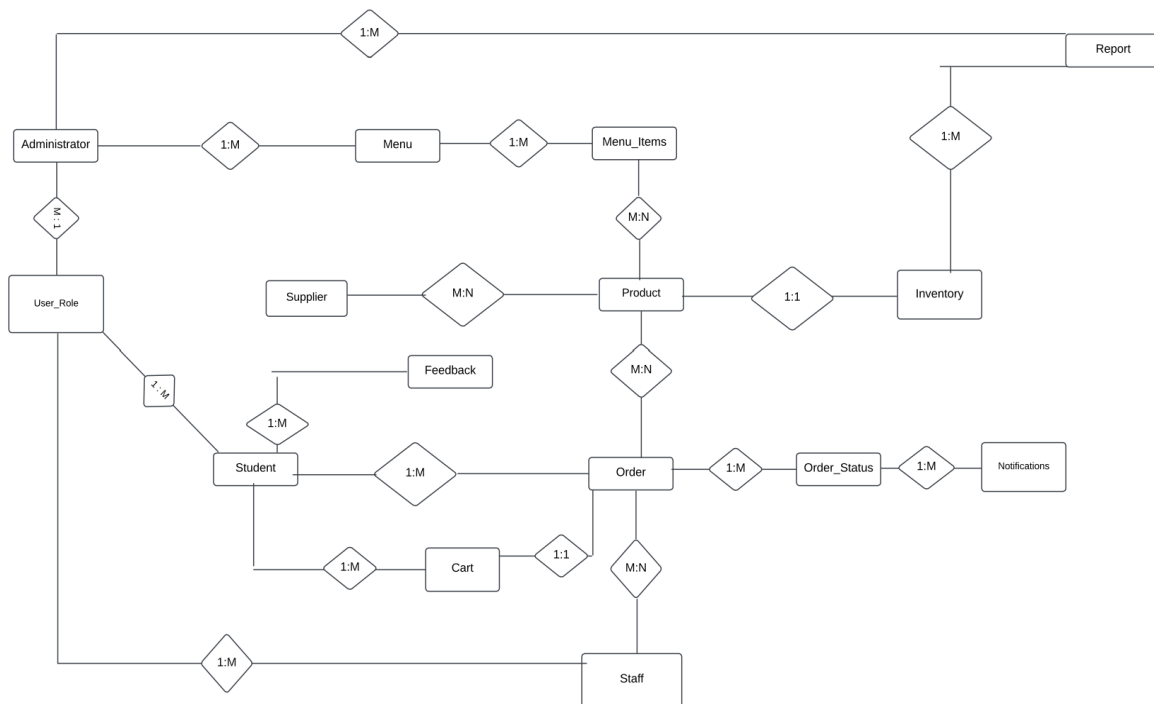


Entities in Database and Data Model

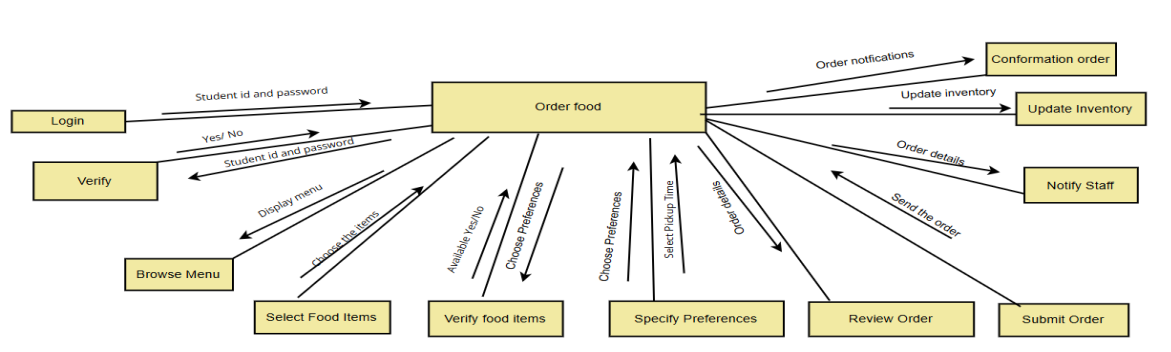
Entities:

Administrator
 Menu
 Menu_Items
 User_Role
 Supplier
 Product
 Inventory
 Feedback
 Student
 Order
 Order_Status
 Cart
 Staff
 Notifications
 Report

Data Model



Architecture Design of The System (Structure Chart and Non Functional Requirements)



Non-Functional Requirements

Operational Requirements:

System Integration: It is a new system. The system should be built to manage student accounts, track inventory, and send notifications, so we can make the order process easy for students and staff.

Portability: our system must work on multiple platforms like web, Android, and iOS.

Performance Requirements:

Speed: our system should be fast. For instance, when the user makes actions, the system should respond to user very fast such as menu loading and order submission within 2 seconds.

Capacity: our system should support up to 500 users during peak hours without problems in performance.

Availability and Reliability: the system should be available almost all the time with very little downtime. The maintenance of the system such as fixing bugs can be done in the night when the restaurant is closed.

Security Requirements:

System Value: we have to protect sensitive data such as student login and inventory details.

Encryption Requirements: all sensitive data such as passwords have to be encrypted.

Access Control: the system should allow different users like students, staff, and administrators to see and use only what they need.

For example:

Students can order food and see their orders.

Staff can view and prepare orders.

Administrators can manage the system, like updating menus or checking reports.

Global and Political Requirements:

Multilingual Support: the system must support multiple languages such as English and French.

Legal Compliance: the system must comply with Canadian privacy laws.

Logic For Each Program

Login Module:

```
Read Student Id
Read Password
Send studentID# to the database
Compare the corresponding password
    If ok {
        Display the message "Login successful."
    }
    Else {
        Display the error message "Login not successful"
    }
```

Menu Browsing Module:

```
Display Menu_Items from database // it could be the name and description of the items.
Read student item_selected
Read student item_preferences
Read the quantity
Save the selected (items_selected, item_preferences) to the Cart
Display the message "Items added to cart successfully."
```

Cart Management Module:

```
Display items_selected in the Cart// list of items chosen
Display update_option // to update the quantity or remove the items
    If (updated option clicked) {
        Read quantity
        Update quantity
        Display the message "Cart updated successfully."
        Update Cart database
    } else {
        Display the error message "No update."
    }

}
```

Order Placement Module:

```
Read Student Id
Display items_selected
    If Cart empty {
        Display error message: "Your cart is empty"
    }
    If else (Item_Availability < Quantity) {
        Display error message: "Item not available in the desired quantity."
    }
    Else {
```

```
    Deduct Items from Inventory database
    Save Order_Record to the database
    Display the message "Order placed successfully. Notification sent."
}
```

Real-Time Order Tracking Module:

```
Read Order_Status from database
Display Order_Status from database
    If Order_Status change {
        Display notification to the student
    } Else {
        No notifications
    }
```

Staff Order Management Module:

```
Read all pending Orders from the database
Display Orders (Order_ID, Items, Quantities)
Save Status_Update to the database
Display the message "Order status updated successfully."
```

Admin Management Module:

```
Display Options //manage menu Items, monitor Inventory
Read option
Display Inventory_Status
Save update to the database
Update Menu
Generate report
Display report to the supplier
```