**UTD Dining and Meal Plan System**

**Phase II. Design and Modeling**

**Course Title: Database Foundations for Business Analytics (BUAN 6320.004)**

**Semester Project Group- 8**

**Group Members:**

Jagdish Umashankar Vishwakarma (juv230000@utdallas.edu)

Uchaansh Lodhi (uxl230001@utdallas.edu)

Sonakshi Handoo (sxh230044@utdallas.edu)

Aditya Chaganti (vxc230016@utdallas.edu)

Utsav Mukesh Maisheri (uxm230000@utdallas.edu)

**Abstract**

This project report outlines developing and implementing a SQL-based Dining and Meal Plan System for The University of Texas at Dallas. The current dining system at UTD lacks a centralized platform, resulting in inefficiencies in meal planning, underutilization of meal plans, and insufficient accommodation of dietary preferences. The proposed system addresses these challenges by providing a comprehensive solution for students and administrators.

The UTD dining and meal plan system, built on SQL, offers a centralized database for managing meal balances, dietary preferences, and transaction histories. This platform enables students to easily track and manage their meal plans, switch plans if desired, and receive personalized food recommendations. Simultaneously, administrators gain tools to manage dining options and meal subscriptions and analyze real-time data to understand student preferences and demand.

The project emphasizes the creation of several database entities, including Students, MealPlans, DiningHalls, Menus, MenuItems, DietaryPreferences, StudentTransactions, Feedback, and Coupons. Each entity is crucial in providing a seamless and personalized dining experience.

Team responsibilities have been equitably distributed, ensuring effective collaboration and timely completion of tasks. The objective is to enhance the efficiency and personalization of the dining experience at UTD, promoting a smoother interaction between students and the university's dining services.

**Content**

1. Executive summary
2. Problem description
3. Conceptual design
4. Relational schema
5. Normalization
6. Conclusion
7. List of Figures
8. List of Tables
9. **Executive Summary**

Our proposed database project is to design and implement a system for the University of Texas at Dallas' (UTD) Dining and Meal Plan to make eating on campus easier and more efficient for its students. Right now, it's tricky for students to know how many meals they have left, change their meal plan, or even find foods that fit their dietary needs. Our project, i.e., UTD dining and meal plan system based on SQL, a type of computer language to manage information, will help solve this problem. With this new system, everything about campus dining will be in one place. Students can easily see and manage their meal balances, pick a different plan, and even get suggestions for finding certain foods. This means no more guessing or wasting time figuring things out. The idea is to make the dining experience at UTD smoother and more personalized for everyone. The system will help administrators manage available dining options, meal plan subscriptions, and student/staff dining transactions.

1. **Problem Description**

The current UTD dining system lacks a centralized platform, leading to inefficiencies in meal planning, underutilization, and inadequate addressing of dietary preferences. Implementing a SQL-based Dining and Meal Plan System offers an opportunity to streamline meal plan management, optimize food preparation based on real-time demand, and cater to diverse dietary needs, enhancing the overall dining experience for students.

1. **Conceptual Design**

Here is the EER diagram generated based on our project description and real-life experiences.

**3.1 EER diagram with all assumptions**

**ER Diagram Link:** [**https://app.diagrams.net/#G12149uuTS4q8384m\_hZavfXBz7Pn4AXER**](https://app.diagrams.net/#G12149uuTS4q8384m_hZavfXBz7Pn4AXER)

**(Click on the link and select “Diagrams.net” option to view the ER diagram clearly )**

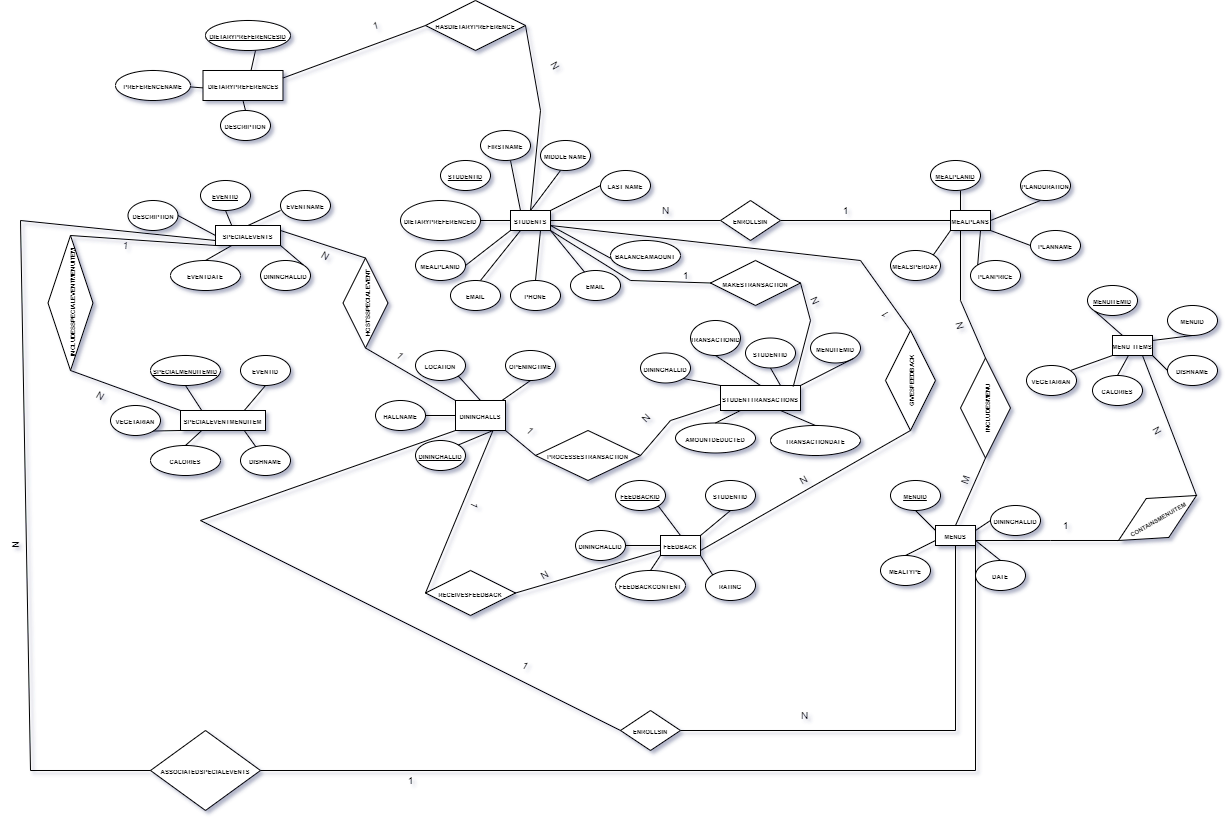
****

Figure 1. EER Diagram

**3.2 (Min, Max) Notation for Relationship**

In this part we discuss the (min, max) notation for several important relationships that exist in our EER diagram. Table 1 clearly specifies how the numerical expression corresponds to the relationship between two entities.

Table 1. Explanation for (Min, Max) Notation

| **Relationship** | **Cardinality** | **Explanation** |
| --- | --- | --- |
| Students - MealPlans | Students(0, 1) to (0, N) MealPlans | (One student can have one meal plan, and one meal plan is associated with many student.) |
| Students - DietaryPreferences | Students(0, 1) to (1, N)DietaryPreferences | (A student may not have a dietary preference, but if they do, they have exactly one dietary preference.  One DietararyPreference is associated with one or many students) |
| Students - StudentTransactions: | Students (1, N) to (1, 1) StudentTransactions | (A student must have at least one transaction but can have multiple transactions.  A transaction is associated with only one student.) |
| Students - Feedback: | Students (1, N) to (1,1) Feedback | (A student must provide at least one feedback but can provide multiple feedback entries.  but a feedback entry is associated with only one student.) |
| MealPlans - Menus: | MealPlans (1, N) to (1, M) Menus | (A meal plan may have one menu, or it can have multiple menus, and a menu can be associated with one or multiple meal plans.) |
| DiningHalls - Menus | DiningHalls (1, N) to (1, 1) Menus | (A dining hall must have at least one menu, but can have multiple menus, and a menu is associated with exactly one dining hall.) |
| Menus - MenuItems: | Menus (1, N) to (1, N) MenuItems | (A menu must have at least one menu item, but can have multiple menu items, and a menu item is associated with one or many menu.) |
| DiningHalls - StudentTransactions: | DiningHalls (1, N) to (1,1) StudentTransactions | (A dining hall must have at least one transaction, but can have multiple transactions, and a transaction is associated with exactly one dining hall.) |
| DiningHalls - Feedback: | DiningHalls (1, N) to (1,1) Feedback | (A dining hall must have at least one feedback entry, but can have multiple entries, and a feedback entry is associated with exactly one dining hall.) |
| DiningHalls - SpecialEvents: | DiningHalls (1, N) to (1,1) SpecialEvents | (A dining hall must have at least one special event, but can have multiple events, and a special event is associated with exactly one dining hall.) |
| SpecialEvents - SpecialEventMenuItems: | SpecialEvents (1, N) to (1, N) SpecialEventMenuItems | (A special event must have at least one menu item, but can have multiple items, and a menu item is associated with one or many special event.) |
| Menus - SpecialEvents: | Menus (0, N) to (0, N) SpecialEvents | (A menu be associated with zero or more special events, and a special event is associated with zero or more menus.) |

1. **Relational Schema**

**4.1 Relational Schema**

Our Relational Schema has no weak entities

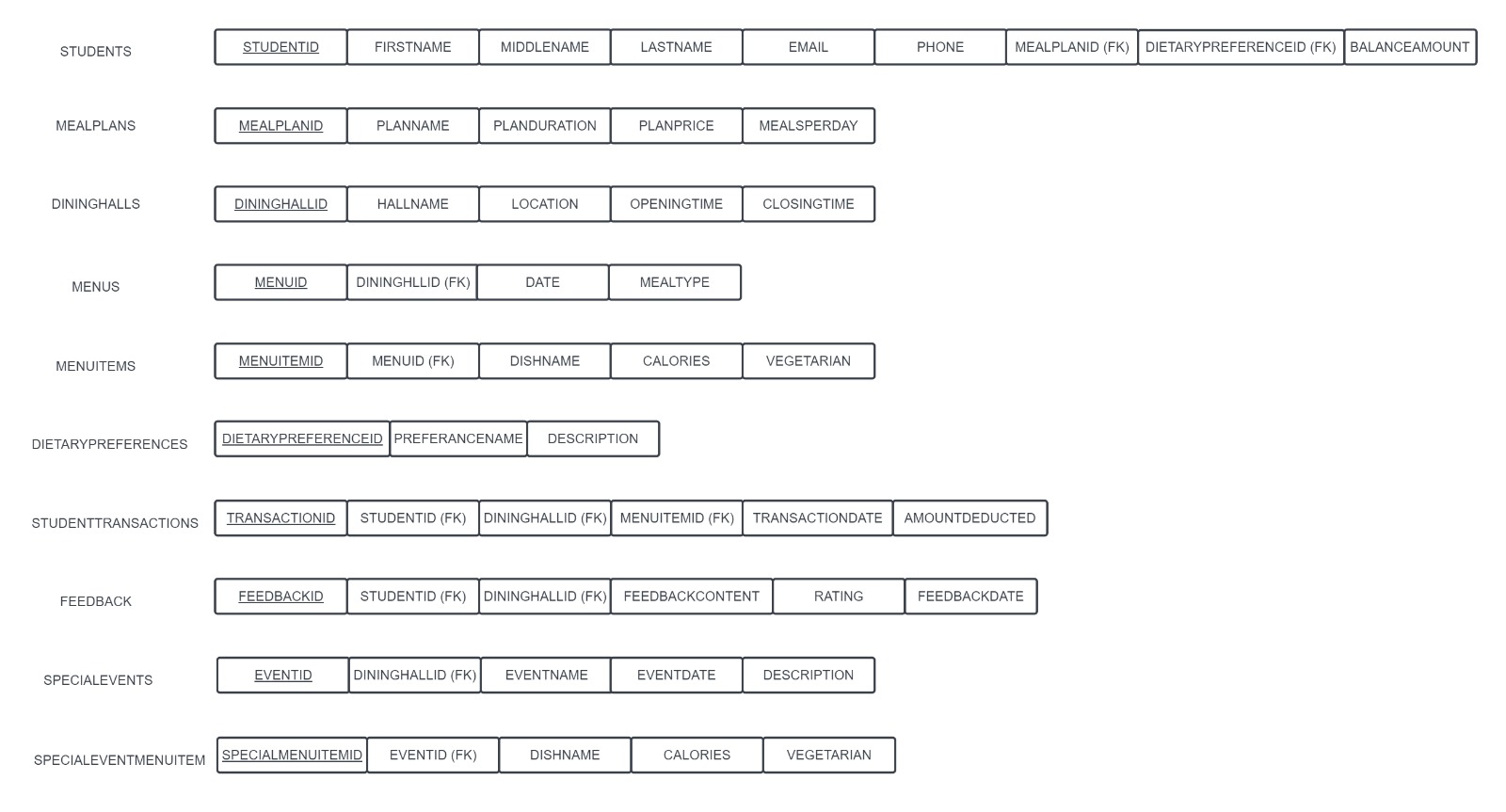
****

Figure 2: Relational Schema

**4.2 Data Format for Every Relation**

Table 2. Data Format for Each Relation

| **Relation Names** | **Attributes** | **Data Type** | **Length (Max) or Size** |
| --- | --- | --- | --- |
| Students | StudentID (PK) | String (Primary Key) | 10 |
|  | FullName | String | 50 |
|  | Email | String | 255 |
|  | Phone | String | 12 |
|  | MealPlanID (FK) | String (Foreign Key) | 10 |
|  | DietaryPreferenceID (FK) | String (Foreign Key) | 10 |
|  | BalanceAmount | Decimal | - |
| MealPlans | MealPlanID (PK) | String (Primary Key) | 10 |
|  | PlanName | String | 50 |
|  | PlanDuration | String | 20 |
|  | PlanPrice | Decimal | - |
|  | MealsPerDay | Integer | - |
| DiningHalls | DiningHallID (PK) | String (Primary Key) | 10 |
|  | HallName | String | 50 |
|  | Location | String | 50 |
|  | OpeningTime | Time | - |
|  | ClosingTime | Time | - |
| Menus | MenuID (PK) | String (Primary Key) | 10 |
|  | DiningHallID (FK) | String (Foreign Key) | 10 |
|  | Date | Date | - |
|  | MealType | String | 20 |
| MenuItems | MenuItemID (PK) | String (Primary Key) | 10 |
|  | MenuID (FK) | String (Foreign Key) | 10 |
|  | DishName | String | 50 |
|  |  |  |  |
|  | Calories | Decimal | - |
|  | IsVegetarian | Boolean | - |
| DietaryPreferences | DietaryPreferenceID (PK) | String (Primary Key) | 10 |
|  | PreferenceName | String | 50 |
|  | Description | String | - |
| StudentTransactions | TransactionID (PK) | String (Primary Key) | 10 |
|  | StudentID (FK) | String (Foreign Key) | 10 |
|  | DiningHallID (FK) | String (Foreign Key) | 10 |
|  | MenuItemID (FK) | String (Foreign Key) | 10 |
|  | TransactionDate | Date | - |
|  | AmountDeducted | Decimal | - |
| Feedback | FeedbackID (PK) | String (Primary Key) | 10 |
|  | StudentID (FK) | String (Foreign Key) | 10 |
|  | DiningHallID (FK) | String (Foreign Key) | 10 |
|  | FeedbackContent | String | - |
|  | Rating | Integer | - |
|  | FeedbackDate | Date | - |
| SpecialEvents | EventID (PK) | String (Primary Key) | 10 |
|  | DiningHallID (FK) | String (Foreign Key) | 10 |
|  | EventName | String | 50 |
|  | EventDate | Date | - |
|  | Description | String | - |
| SpecialEventMenuItems | SpecialMenuItemID (PK) | String (Primary Key) | 10 |
|  | EventID (FK) | String (Foreign Key) | 10 |
|  | DishName | String | 50 |
|  |  |  |  |
|  | Calories | Decimal | - |
|  | IsVegetarian | Boolean | - |

**5. Normalization**

In this part, we apply the principles of normalization to ensure all the tables conform to 3NF. To do this, we document all functional dependencies and indicate how the normalization is performed.

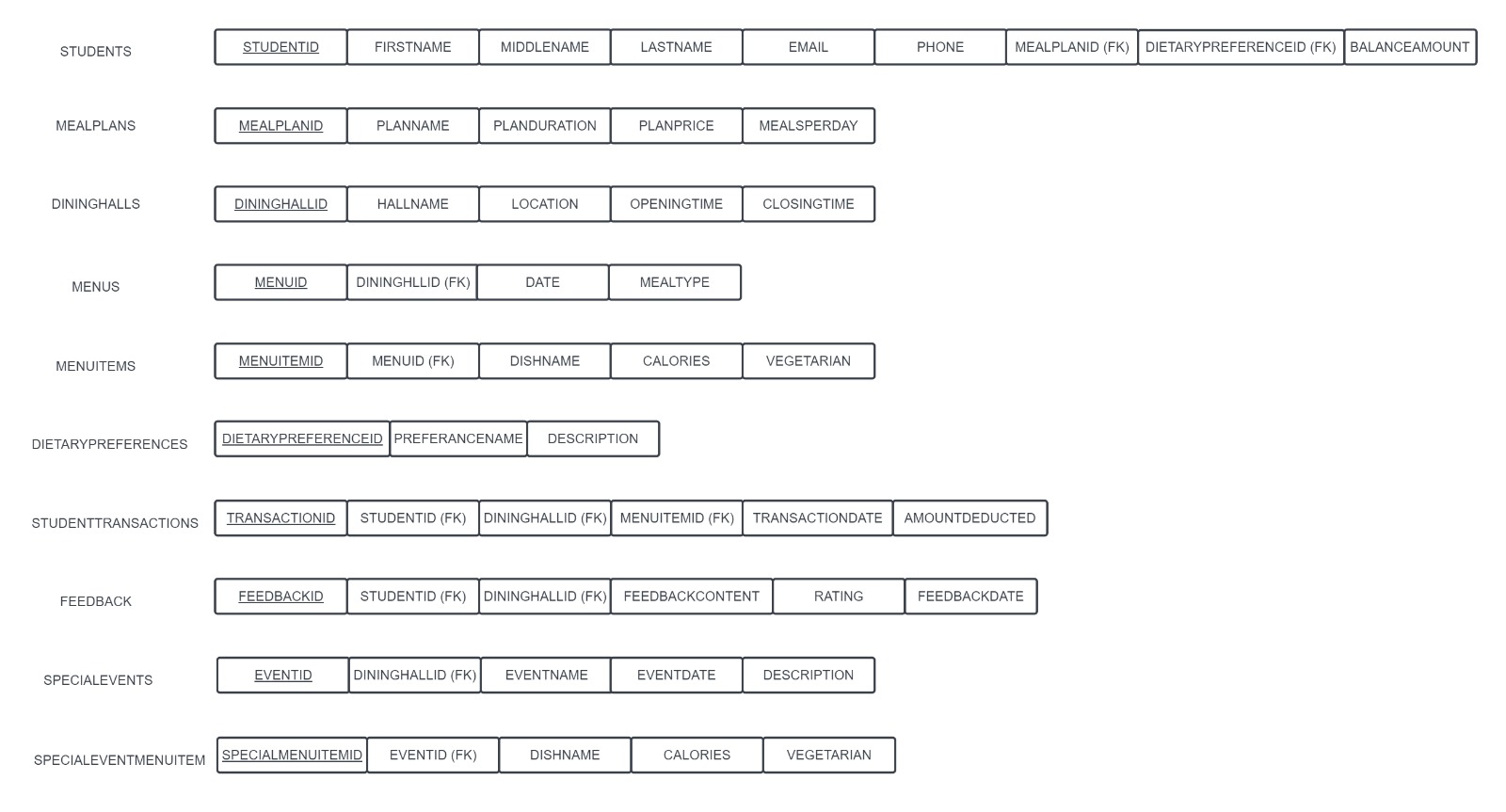
****

Figure 3: Normalized Relational Schema

**6. Conclusion**

In this report, we discuss and design the relational schema of the UTD Dining Database. Our EER diagram and its associated relational schema show the conceptual and logical designs of the system. We also define data types and formats for each attribute in the relation. The next step is to implement this database. We may change some designs in the future due to practical difficulties and other requirements.

**7. List of Figures**

Figure 1: EER Diagram

Figure 2: Relational Schema  
Figure 3: Normalized Relational Schema

**8. List of Tables**

Table 1. Explanation for (Min, Max) Notation

Table 2. Data Format for Each Relation