PROJECT 0502 03 PROPOSAL

1. Project analysis and design

Foster Assistant:

Foster Assistant is an intelligent visual assistant product, which requires a database to store the operational data related to its users and their actions performed. The product is capable of taking notes, setting reminders and making task lists for the user. These aspects have been mapped to the proposed database design for proper functioning of the product.

Every user has a unique identifier assigned to it, and requires to store his or her own full name (consisting of two fields - first name and last name), date of birth, gender, email address, phone number and location (consisting of two fields - city and country). The users stored in the database must own one or more of these devices (product). Each device is a product with unique identifier, name, serial number and manufacturing date and it is owned by exactly one user. The user may create many tasks, through voice recognition technology. Each task has its own unique identifier, description, status (whether it is new, pending or done), and priority (whether it is low, medium or high in urgency). One task may include a note or a reminder or both. Every note has a unique identifier and text associated with it as description. Every reminder has a unique identifier, description, repeat option (whether it is to be repeated daily, weekly or monthly), date and time. Apart from these actions, the user may also request queries that require the product to map these commands to external systems. Every query is a mapping of command to its response system. For example, if a user requests to play a song, then this command is mapped to Spotify server as the response. A user may request one or many queries and similarly, one query may be requested by one or many users.

ER Schema:

Entities, Attributes and Primary Keys

User(**userId**, userName, -userFirstName, -userLastName, userDOB, userGender, userEmail, userPhone, userLocation, -userCity, -userCountry)
Device (**dvcId**, dvcName, dvcSerialNumber, dvcManufactureDate)
Task(**taskId**, taskDescription, taskStatus, taskPriority)
Reminder(**remId**, remDescription, remDate, remTime, remRepeat)
Note(**noteId**, noteDescription)
Query(**queryId**, queryCommand, queryResponse)

Relationships, Attributes, Degrees, Participating Entities and Constraints

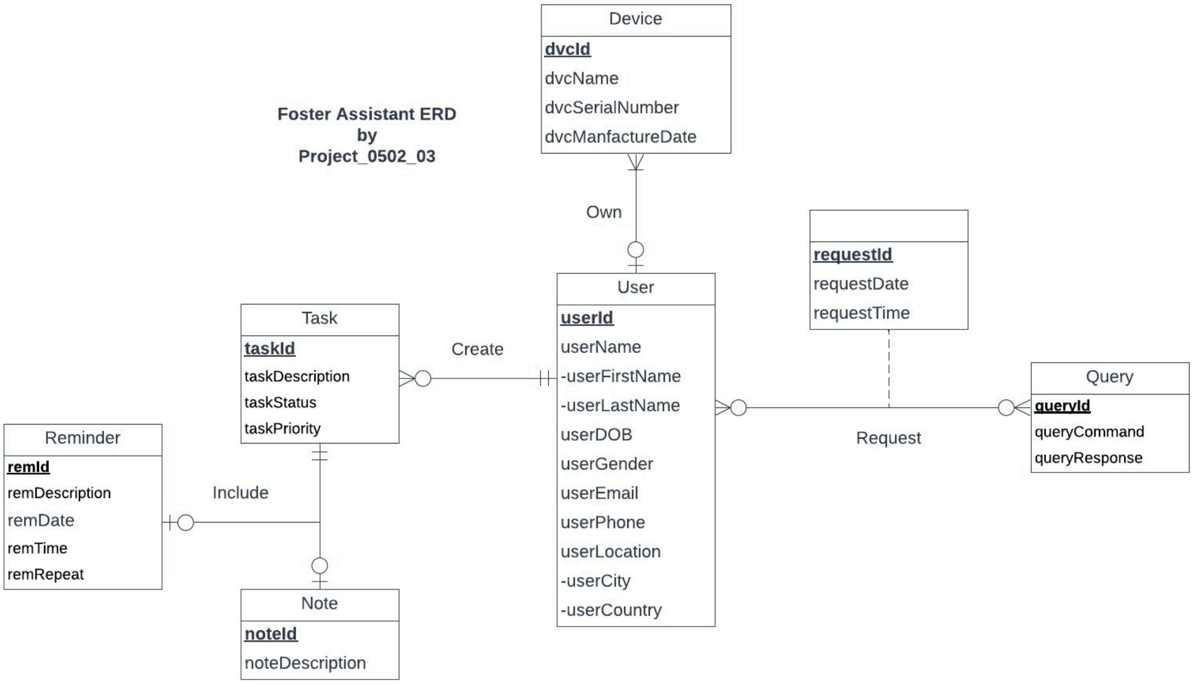
Request : binary relationship
1 User to 0 or more Query
1 Query to 0 or more User

Create : binary relationship
1 User to 0 or more Task
1 Task to 1 User

Include : ternary relationship
1 Task and 1 Note to 0 or 1 Reminder
1 Reminder and 1 Note to 1 Task
1 Task and 1 Reminder to 0 or 1 Note

Own : binary relationship
1 Device to 0 or 1 User
1 User to 1 or more Device

ER Diagram:



2. Project proposal and data

Mission Statement:

We aim to create a better experience for the user by analyzing the database of Foster Assistant. By inspecting the user's requests for the virtual assistant, we seek to understand the users' demands and their behavior, hence, maintain the database accordingly. Thus, creating a personalized product for the user as well as sustaining profits and industry relationships.

Mission Objectives:

The primary objectives of the database are:

1. Finding what type of query, task, and reminder is most performed by the users to understand user behavior and create a smarter product.
2. Streamline relationships with third-party service providers based on user behavior to add, upgrade or remove connections to external systems so that contracts and finances can be managed.
3. To get the most inactive users so that promotional emails can be sent to encourage them to use the software.

4. To analyze the type of tasks and their respective priorities created by the user to personalize the task planner on the device.
5. Understand and analyze locations of users to formulate an expansion strategy by understanding the scope outside the United States and also draw plans on a marketing budget for the cities in the United States.

Relations:

User(**userId**, userFirstName, userLastName, userDOB, userGender, userEmail, userPhone, userCity, userCountry)
Device(**dvcId**, dvcName, dvcSerialNumber, dvcManufactureDate, *userId*)
Task(**taskId**, taskDescription, taskStatus, taskPriority, *userId*)
Query(**queryId**, queryCommand, queryResponse)
Request(**requestId**, *userId, queryId,* requestDate, requestTime)
Note(**noteId**, noteDescription)
Reminder(**remId**, remDescription, remDate, remTime, remRepeat)
Include(***taskId****, noteId, remId*)

Determine functional dependencies and verify normalization to 3NF:

1. Since there are no multi-valued attributes, the relations are in 1NF form
2. Since there are no partial dependencies, the relations are in 2NF form
3. Since there are no transitive dependencies, the relations are in 3NF form

Hence, our database relations have been verified to 3NF form.

Business Rules:

[R1]: When a user owns a device, the user cannot be deleted from the database
[R2]: When a user owns a device, the user cannot be updated in the database
[R3]: When a user creates a task, the user cannot be deleted from the database
[R4]: When a user creates a task, the user cannot be updated in the database
[R5]: When a user requests a query, the user cannot be deleted from the database
[R6]: When a user requests a query, the user cannot be updated in the database
[R7]: When a query is requested by a user, the query cannot be deleted from the database
[R8]: When a query is requested by a user, the query cannot be updated in the database
[R9]: When a task is deleted from the database, all notes and reminders included in the task should be deleted in the database
[R10]: When task information is changed in the database, the corresponding task information associated to a reminder and a note, should be updated in the database accordingly.
[R11]: When a note is deleted from the database, the note information associated with the corresponding task and reminder should be set to null.
[R12]: When note information is changed in the database, the corresponding note information associated to a task and a reminder, should be updated in the database accordingly.

[R13]: When a reminder is deleted from the database, the reminder information associated with the corresponding task and note should be set to null.
[R14]: When reminder information is changed in the database, the corresponding reminder information associated to a task and a note, should be updated in the database accordingly.

Referential Integrity:

Relation	Foreign Key	Base Relation	Primary Key	Business Rule	Constraint: ON DELETE	Business Rule	Constraint ON UPDATE
Device	userId	User	userId	R1	NO ACTION	R2	NO ACTION
Task	userId	User	userId	R3	NO ACTION	R4	NO ACTION
Request	userId	User	userId	R5	NO ACTION	R6	NO ACTION
Request	queryId	Query	queryId	R7	NO ACTION	R8	NO ACTION
Include	taskId	Task	taskId	R9	CASCADE	R10	CASCADE
Include	noteId	Note	noteId	R11	SET NULL	R12	CASCADE
Include	remId	Reminder	remId	R13	SET NULL	R14	CASCADE

Sample Data:

User(**userId**, userFirstName, userLastName, userDOB, userGender, userEmail, userPhone, userCity, userCountry)

userId	userFirstName	userLastName	userDOB	userGender	userEmail	userPhone	userCity	userCountry
U10000001	David	Brantly	12-09-1986	Male	dbrant@gmail.com	2408543777	College Park	United States

Device(**dvcId**, dvcName, dvcSerialNumber, dvcManufactureDate, *userId*)

dvcId	dvcName	dvcSerialNumber	dvcManufactureDate	userId
D0000001	David’s Foster	KL859468	03-02-2005	U10000001

Task(**taskId**, taskDescription, taskStatus, taskPriority, *userId*)

taskId	taskDescription	taskStatus	taskPriority	userId
TK8567	Get Internship	Pending	High	U10000001

Query(**queryId**, queryCommand, queryResponse)

queryId	queryCommand	queryResponse
Q1000001	Play Song	PUT https://api.spotify.com/v1/me/player/play
Q1000002	Pause Song	PUT https://api.spotify.com/v1/me/player/pause

Request(**requestId**, *userId, queryId,* requestDate, requestTime)

requestId	userId	queryId	requestDate	requestTime
R1000001	U10000001	Q1000001	09-07-2022	11:00:00

Note(**noteId**, noteDescription)

noteId	noteDescription
N12209	Find present employees and ask for referrals

Reminder(**remId**, remDescription, remDate, remTime, remRepeat)

remId	remDescription	remDate	remTime	remRepeat

RM100001	Call Ron	11-10-2022	13:00:00	Daily

Include(***taskId***, *noteId, remId*)

taskId	noteId	remId
TK8567	N12209	RM100001