



Neumann János Egyetem
Műszaki és Informatikai
Kar

Smart Dormitory Access Control System

Database Plan

Team
FreeCredit

IT Project Work
2025

1. Requirement Catalog and Entity-Relationship Diagram

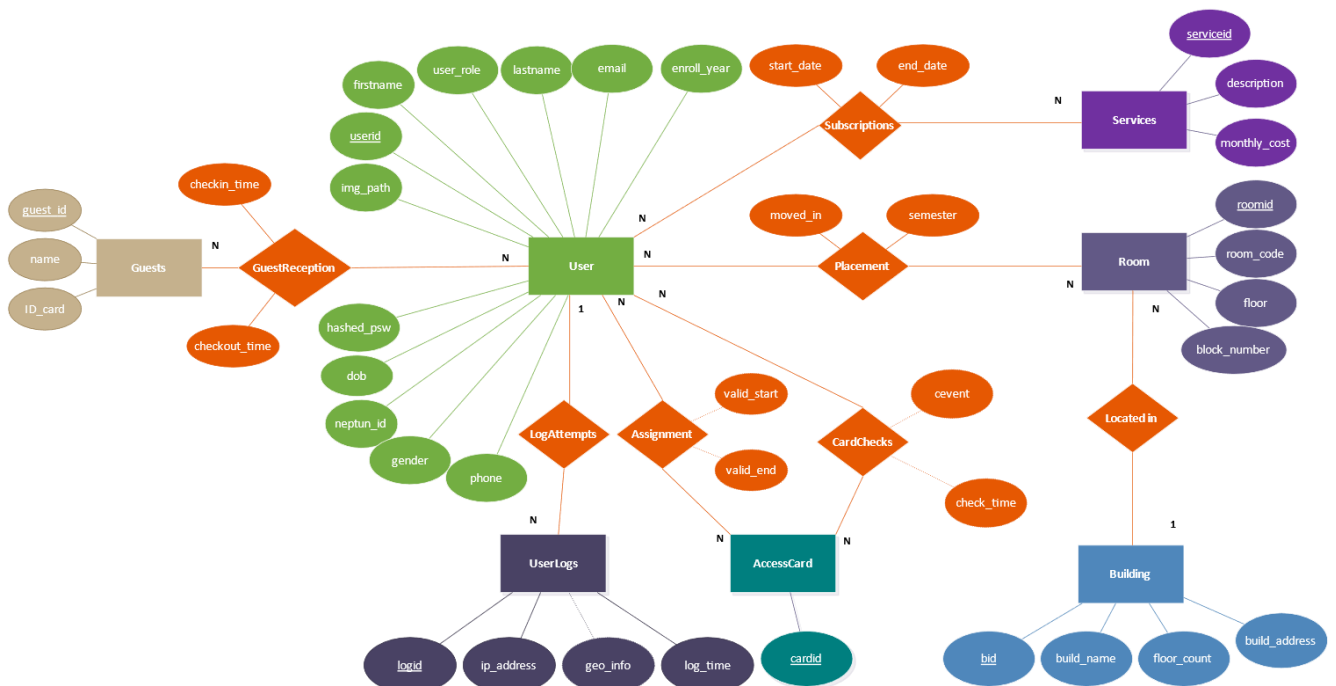
As part of our project, we aim to develop an automated dormitory access control and monitoring system that simplifies student entry using NFC cards. At the same time, it provides valuable insights for dormitory staff by tracking the number of students currently present and maintaining room assignments efficiently.

Our design process began with analyzing the data storage requirements necessary for these functions, following the principles of relational database modeling.

Key data managed and stored by the system:

- Student and staff information
- Card data and logs
- Room and building details
- Guest Reception (*future planning*)
- Service Inventory (*future planning*)

Entity-Relationship Diagram:



2. Entity-Relationship Diagram Mapping

USER(userid, firstname, lastname, email, phone, gender, hashed_psw, img_path, user_role, dob, neptun_id, enroll_year)

ROOM(roomid, room_code, floor, block_number, *bid*)

ACCESSCARD(cardid)

BUILDING(bid, build_name, floor_count, build_address)

USERLOGS(logid, ip_address, geo_info, log_time, *userid*)

SERVICES(*serviceid*, description, monthly_cost)

GUESTS(*guestid*, name, ID_card)

SUBSCRIPTIONS(*userid*, *serviceid*, start_date, end_date)

PLACEMENT(*userid*, *roomid*, semester, moved_in)

CARDCHECKS(*cardid*, *userid*, cevent, check_time)

ASSIGNMENT(*userid*, *cardid*, valid_start, valid_end)

GUESTRECEPTION(*guestid*, *userid*, checkin_time, checkout_time)

3. Normalisation

1NF: None of the tables violate the First Normal Form, as each table has a primary key and does not contain composite attributes.

2NF: The tables comply with the Second Normal Form, as every non-key attribute fully depends on the table's primary key.

3NF: The $\{ip_address\} \rightarrow \{geo_info\}$ transitive dependency in the USERLOGS table violates the Third Normal Form (3NF), so the geo information must be stored in a separate table.

The $\{description\} \rightarrow \{monthly_cost\}$ transitive dependency in the SERVICES table violates the Third Normal Form (3NF), so cost information also must be stored separately.

Final Data Tables:

USER(*userid*, firstname, lastname, email, phone, gender, hashed_psw, img_path, user_role, dob, neptun_id, enroll_year)

ROOM(*roomid*, room_code, floor, block_number, *bid*)

ACCESSCARD(*cardid*)

BUILDING(*bid*, build_name, floor_count, build_address)

USERLOGS(*logid*, ip_address, log_time, *userid*)

SERVICES(*serviceid*, description)

GUESTS(*guestid*, name, ID_card)

SUBSCRIPTIONS(*userid*, *serviceid*, start_date, end_date)

PLACEMENT(*userid*, *roomid*, *semester*, *moved_in*)

CARDCHECKS(*cardid*, *userid*, *cevent*, *check_time*)

ASSIGNMENT(*userid*, *cardid*, *valid_start*, *valid_end*)

GUESTRECEPTION(*guestid*, *userid*, *checkin_time*, *checkout_time*)

GEO_INFO(*ip_address*, *geo_info*)

PRICES(*description*, *monthly_cost*)

4. Data Types and Ranges

USER(*userid*, *firstname*, *lastname*, *email*, *phone*, *gender*, *hashed_psw*, *img_path*,
user_role, *dob*, *neptun_id*, *enroll_year*)

dom(*userid*) = {auto_increment, unsigned integer}

dom(*firstname*) = {string, max 25}

dom(*lastname*) = {string, max 25}

dom(*email*) = {string, max 50}

dom(*phone*) = {string, max 15}

dom(*gender*) = {enum, („male”, „female”)}

dom(*hashed_psw*) = {string, max 255}

dom(*img_path*) = {string, max 255}

dom(*user_role*) = {enum, („student”, „staff”)}

dom(*dob*) = {time, valid}

dom(*neptun_id*) = {string, max 15}

dom(*enroll_year*) = {string, max 25}

GUESTS(*guestid*, *name*, *ID_card*)

dom(*guest_id*) = {auto_increment, unsigned integer}

dom(*name*) = {string, max 50}

dom(*ID_card*) = {string, max 50}

SERVICES(*serviceid*, *description*)

dom(*serviceid*) = {auto_increment, unsigned integer}

dom(*description*) = {string, max 120}

ACCESSCARD(*cardid*)

dom(*cardid*) = {string, max 255}

BUILDING(**bid**, build_name, floor_count, build_address)

dom(bid) = {auto_increment, unsigned integer}

dom(build_name) = {string, max 255}

dom(floor_count) = {unsigned integer}

dom(build_address) = {string, max 255}

ROOM(**roomid**, room_code, floor, block_number, *bid*)

dom(roomid) = {auto_increment, unsigned integer}

dom(room_code) = {string, max 25}

dom(floor) = {unsigned integer}

dom(block_number) = {unsigned integer}

FOREIGN KEY(*BUILDING.bid*)

USERLOGS(**logid**, ip_address, log_time, *userid*)

dom(logid) = {auto_increment, unsigned integer}

dom(ip_address) = {string, max 32}

dom(log_time) = {datetime, valid}

FOREIGN KEY(*USER.userid*)

PLACEMENT(*userid*, **roomid**, semester, moved_in)

dom(semester) = {string, max 25}

dom(moved_in) = {boolean}

FOREIGN KEY(*USER.userid*)

FOREIGN KEY(*ROOM.roomid*)

SUBSCRIPTIONS(*userid*, **serviceid**, start_date, end_date)

dom(start_date) = {datetime, valid}

dom(end_date) = {datetime, valid}

FOREIGN KEY(*USER.userid*)

FOREIGN KEY(*SERVICES.serviceid*)

CARDCHECKS(**cardid**, *userid*, cevent, check_time)

dom(cevent) = {string, max 50}

dom(check_time) = {datetime, valid}

FOREIGN KEY(*ACCESSCARD.cardid*)

FOREIGN KEY(*USER.userid*)

GEO_INFO(ip_address, geo_info)

dom(geo_info) = {string, max 255}

FOREIGN KEY(*USERLOGS.ip_address*)

ASSIGNMENT(userid, cardid, valid_start, valid_end)

dom(valid_start) = {datetime, valid}

dom(valid_end) = {datetime, valid}

FOREIGN KEY(*USER.userid*)

FOREIGN KEY(*ACCESSCARD.cardid*)

GUESTRECEPTION(guestid, userid, checkin_time, checkout_time)

dom(roomid) = {datetime, valid}

dom(room_code) = {datetime, valid}

FOREIGN KEY(*USER.userid*)

FOREIGN KEY(*GUESTS.guest_id*)

PRICES(description, monthly_cost)

dom(monthly_cost) = {number}

FOREIGN KEY(*SERVICES.description*)