

THE ANDERSONS
LST-B

Analysis and design report

Karlís KalnaKarklis
Klaudija Rackauskaite
Mehmet Gormez
Muhammed Azaam Ali
Victor Chukwuma
Rune Lemmens



CONTENTS

1.INTRODUCTION	3
2.REQUIREMENTS ANALYSIS	4
2.1. Functional requirements	5
2.1.1.Use Case Diagram	5
2.1.2.Use Cases	6
2.1.2.1. Login	6
2.1.2.2. Update account settings	6
2.1.2.3. Change password	6
2.1.2.4. Recover Password	6
2.1.2.5. Join planned meal	7
2.1.2.6. Print schedule	7
2.1.2.7. Manage trips	8
2.1.2.8. Manage Checkpoints	8
2.1.2.9. Confirm checkpoint	8
2.1.2.10. Manage personal allergies	9
2.1.2.11. Manage personal tasks	9
2.1.2.12. Request Collaboration	10
2.1.2.13. Manage personal receipts	10
2.1.2.14. Print receipts	11
2.1.2.15. Manage availability	11
2.1.2.16. Manage planned meal	11
2.1.2.17. Manage staff receipts	12
2.1.2.18. Manage staff tasks	12
2.1.2.19. Manage account	13
2.1.2.20. Send task reminder	13
2.2. Non-functional requirements	13
2.2.1.Usability & Accessibility	13
2.2.2.Performance	13
2.2.3.Reliability	13
2.2.4.Security & Privacy	13
3.DATA MODEL	15
4.PRIORITY BY FUNCTIONALITY	25

1. Introduction

This report describes the analysis and design of a household management application developed for the Andersons as part of the Skills Integration Lab 2. The goal of this project is to better understand the needs of the household and translate them into clear system requirements and design choices. Currently, tasks, schedules, meals, trips, and expenses are not managed in a centralised way, which can lead to confusion and lack of overview among staff and family members.

In this document, the functional and non-functional requirements of the system are analysed in detail. Use cases are defined to explain how different users interact with the application, and a data model is created to show how information is structured and related.

Finally, priorities are assigned to the functionalities using the MoSCoW method to clarify which features are the most important.

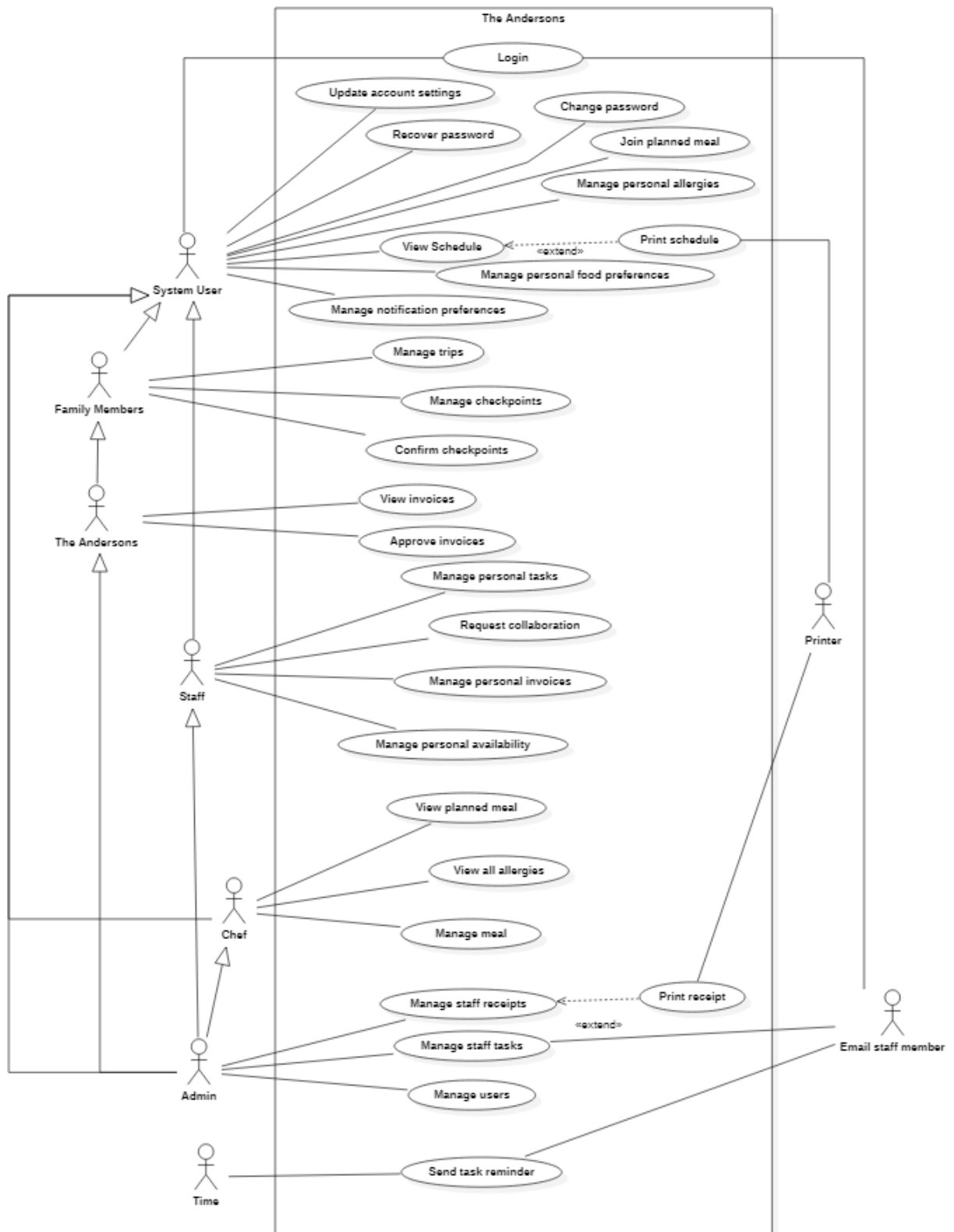
This report serves as a foundation for the further development and implementation of the application.

2. Requirements analysis

This section outlines the specific functional requirements needed to meet the objectives defined for the Andersons' household management application. The core issue to be solved is the lack of a unified overview and clear ownership of the tasks among the staff.

The requirements focus on creating a single source for scheduling, managing resources, tracking finances, and accounting for the specific needs of family members, such as allergies and the grandson's need to know who is in the house.

2.1. Functional requirements



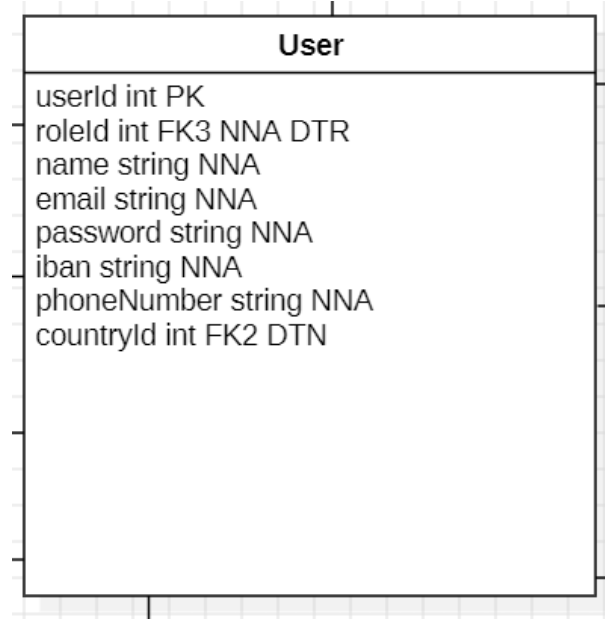
2.1.1. Use Cases

2.1.1.1. Login

Functionality: As a System User, I can login.

Preconditions: Actor is logged in as the System User

Data Model View:

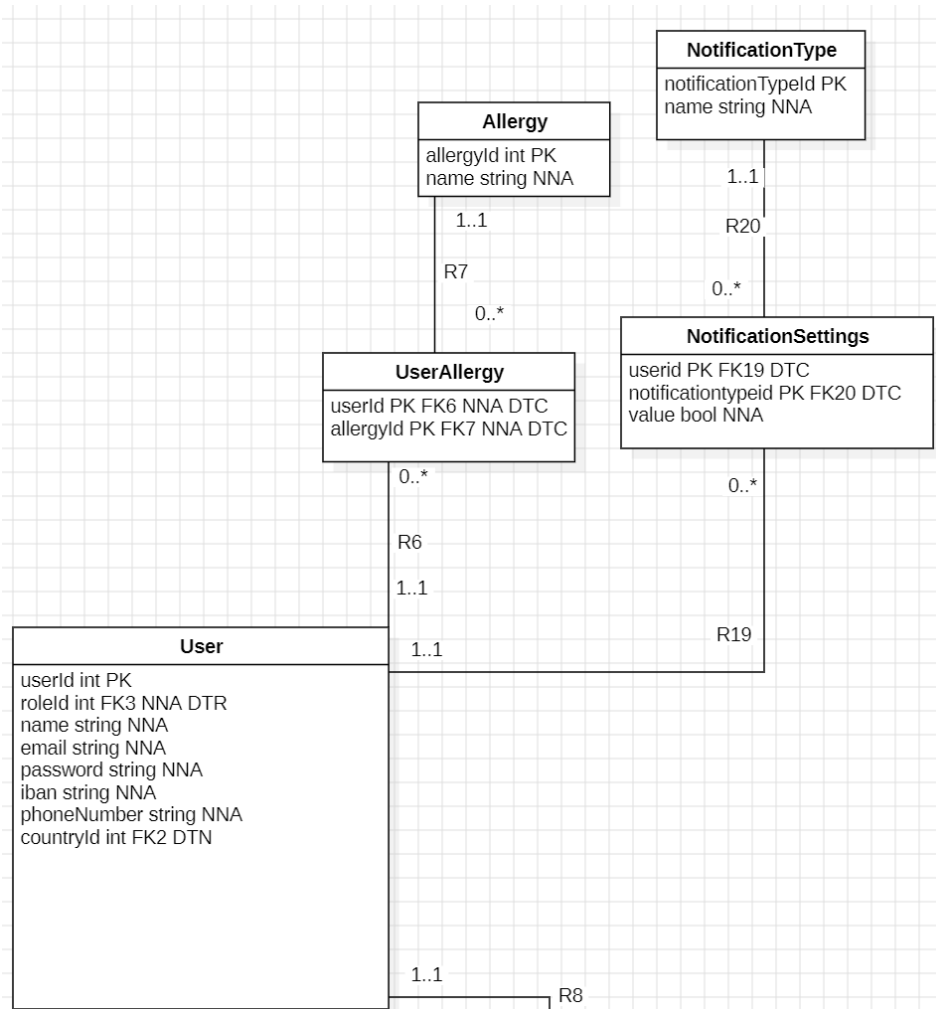


2.1.1.2. Update account settings

Functionality: As a System User, I can update my account settings.

Preconditions: Actor is logged in as the System User

Data Model View:

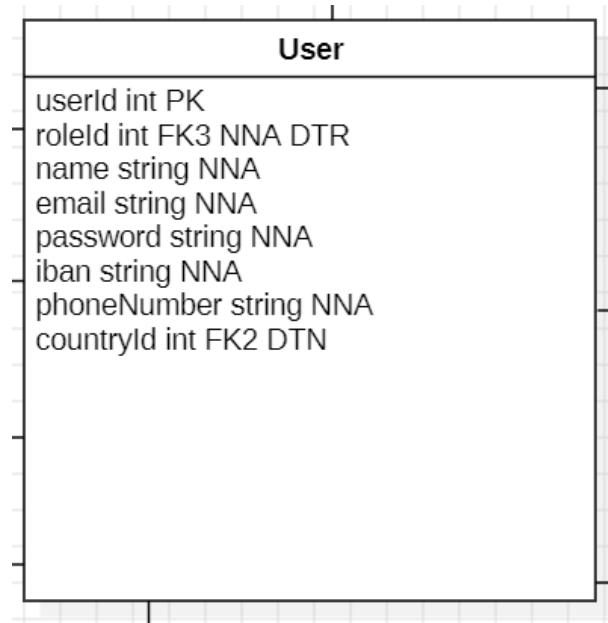


2.1.1.3. Change password

Functionality: As System User, I can change password.

Preconditions: Actor is logged in as the System User

Data Model View:

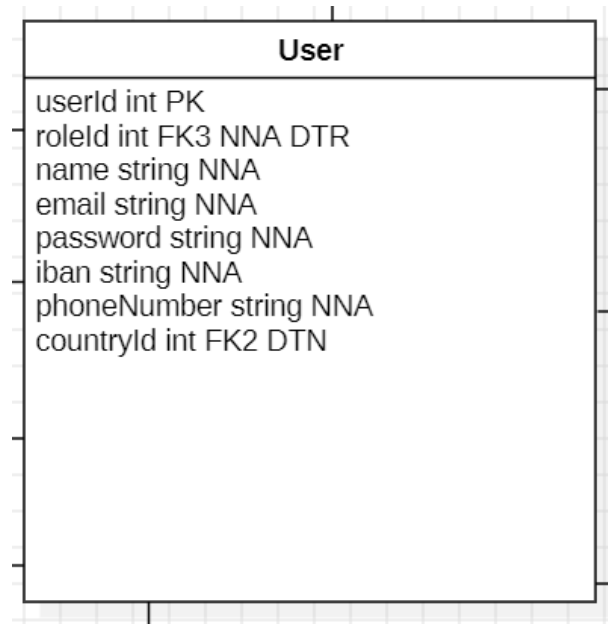


2.1.1.4. Recover Password

Functionality: As a System User, I can recover my password.

Preconditions: Actor is logged in as the System User

Data Model View:

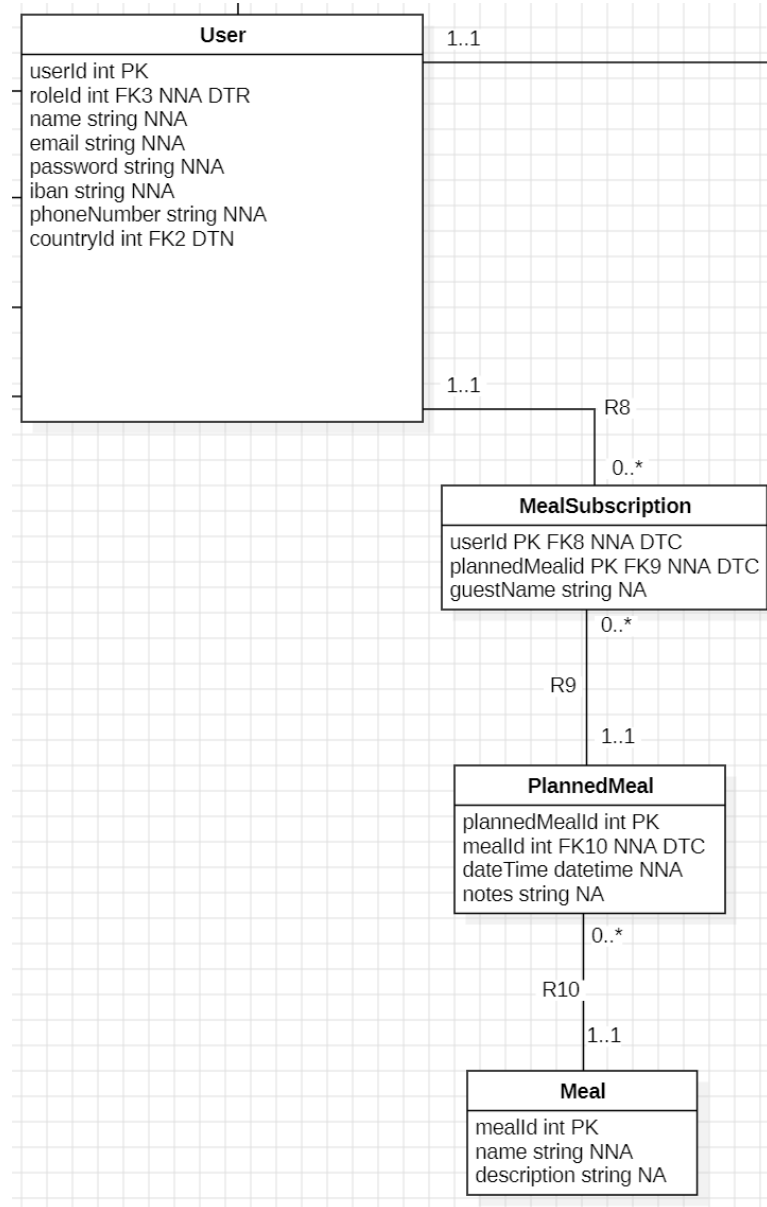


2.1.1.5. Join planned meal

Functionality: As a System User, I can join a planned meal.

Preconditions: Actor is logged in as the System User

Data Model View:

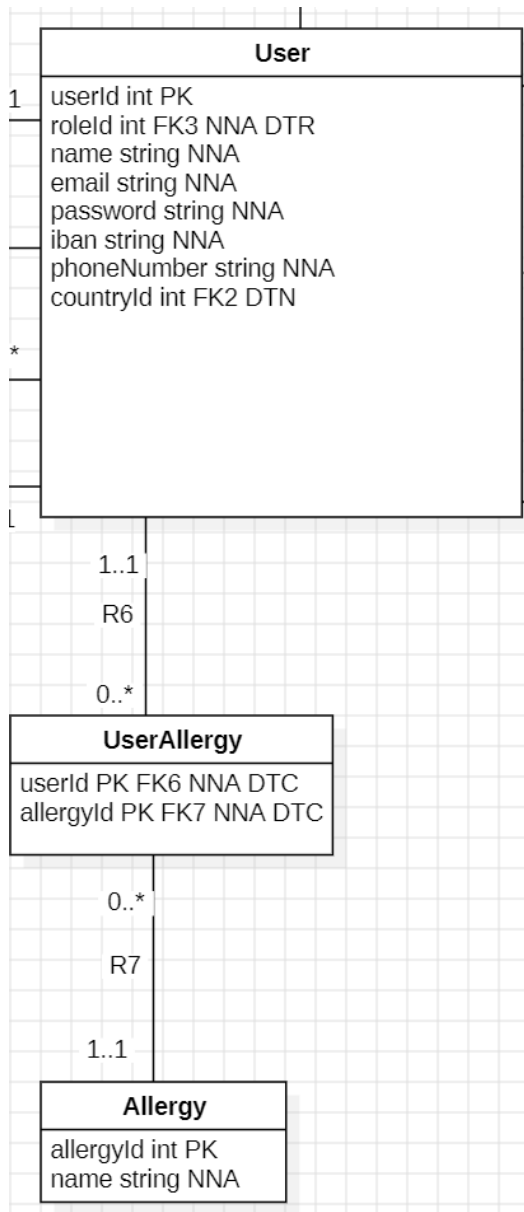


2.1.1.6. Manage personal allergies

Functionality: As a System User, I can manage my personal allergies.

Preconditions: Actor is logged in as the System User

Data Model View:

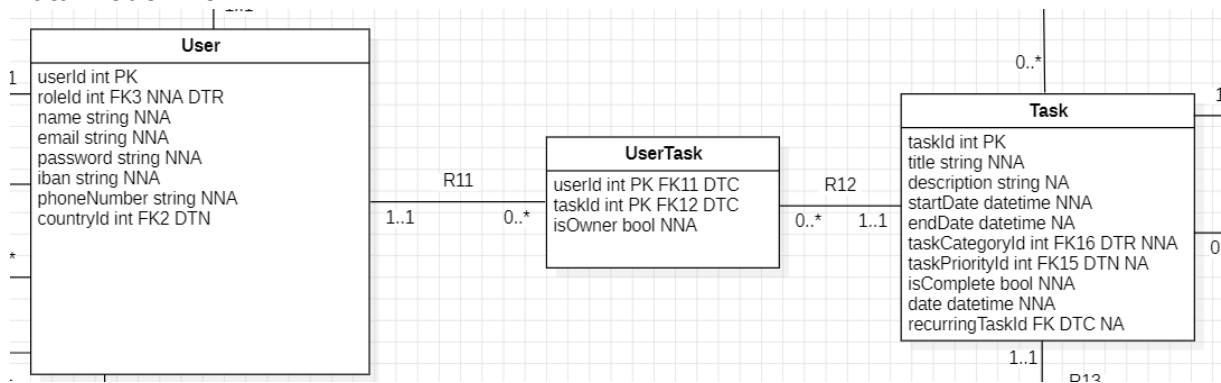


2.1.1.7. View Schedule

Functionality: As a System User, I can view the whole schedule.

Preconditions: Actor is logged in as the System User

Data Model View:

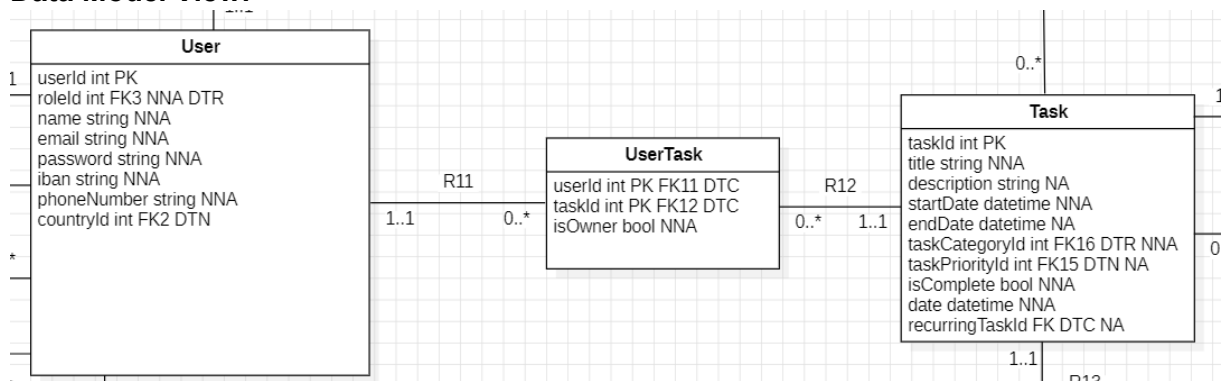


2.1.1.8. Print schedule

Functionality: As a System User, I can print the my schedule.

Preconditions: Actor is logged in as the System User

Data Model View:

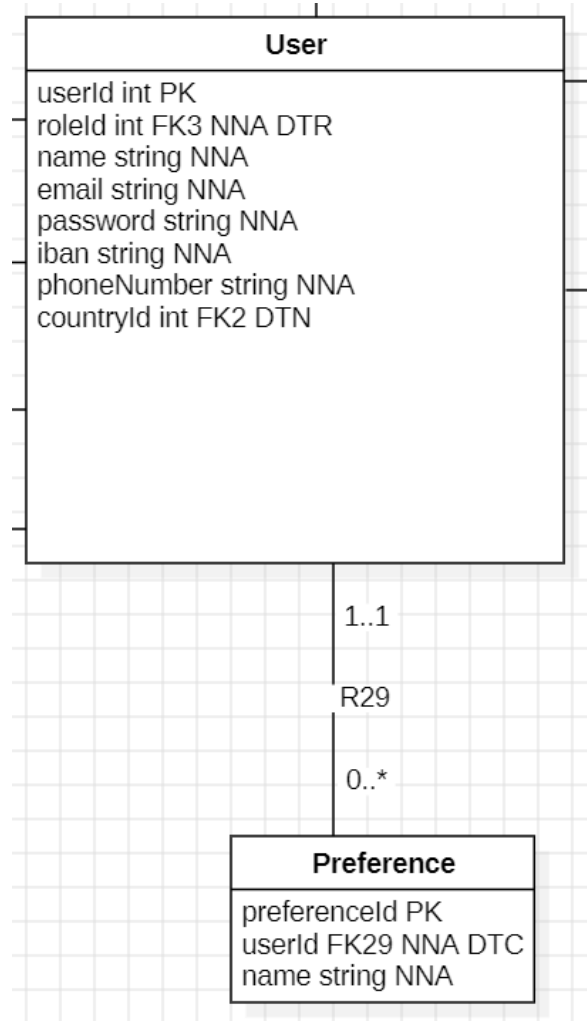


2.1.1.9. Manage personal food preferences

Functionality: As a System User, I can manage my food preferences.

Preconditions: Actor is logged in as the System User

Data Model View:

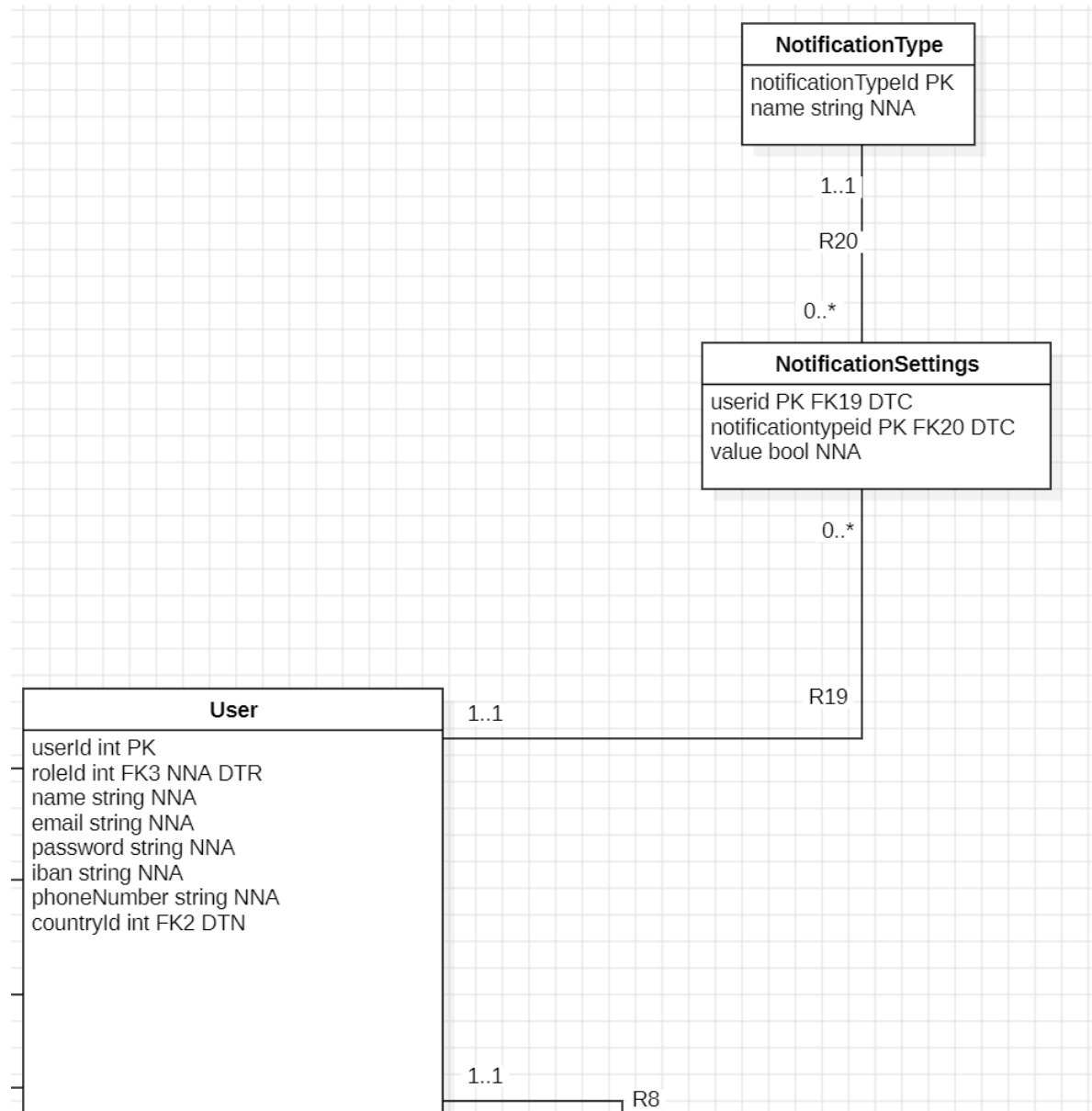


2.1.1.10. Manage notification preferences

Functionality: As a System User, I can manage notification preferences.

Preconditions: Actor is logged in as the System User

Data Model View:

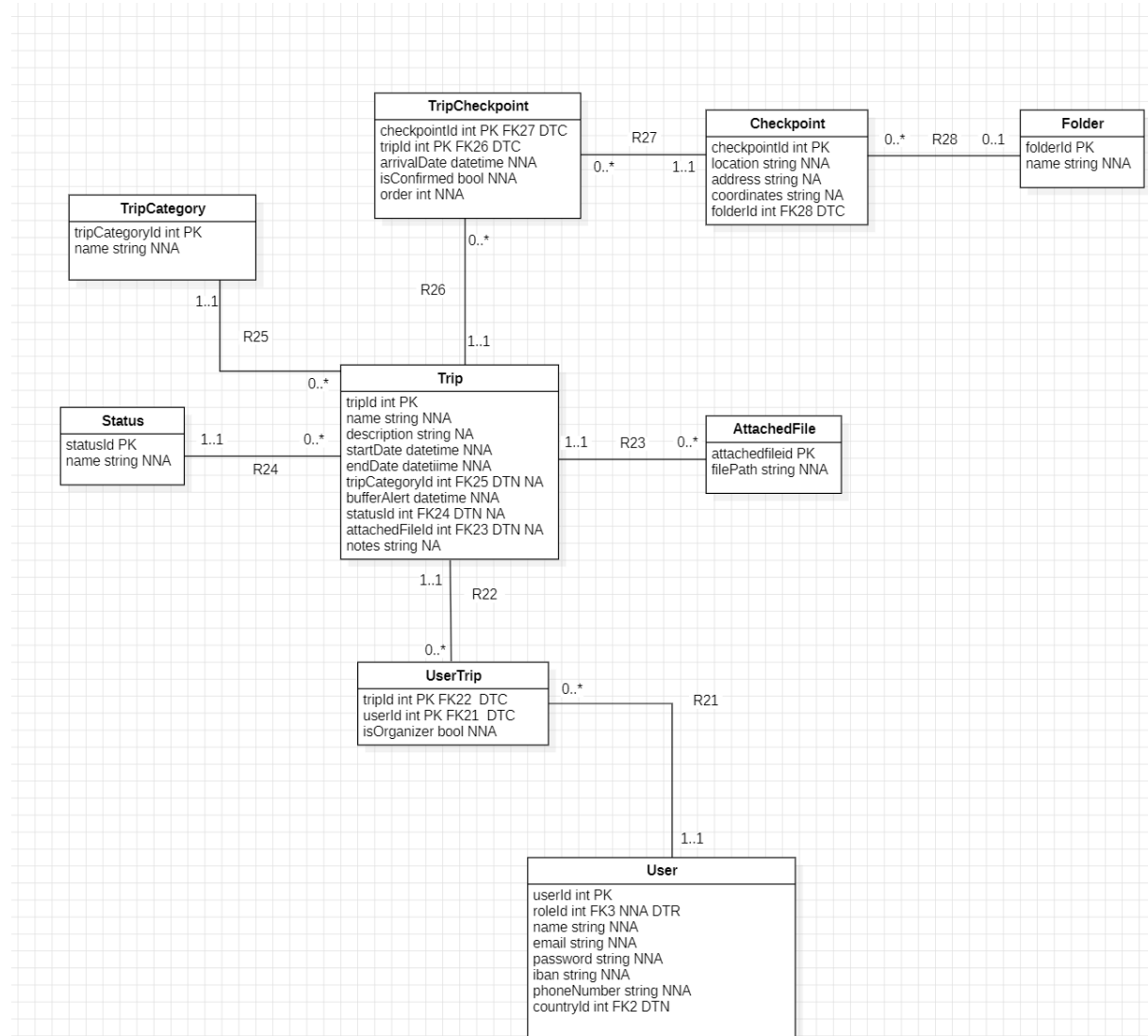


2.1.1.11. Manage trips

Functionality: As a Family member, I can view trips.

Preconditions: Actor is logged in as the System User

Data Model View:

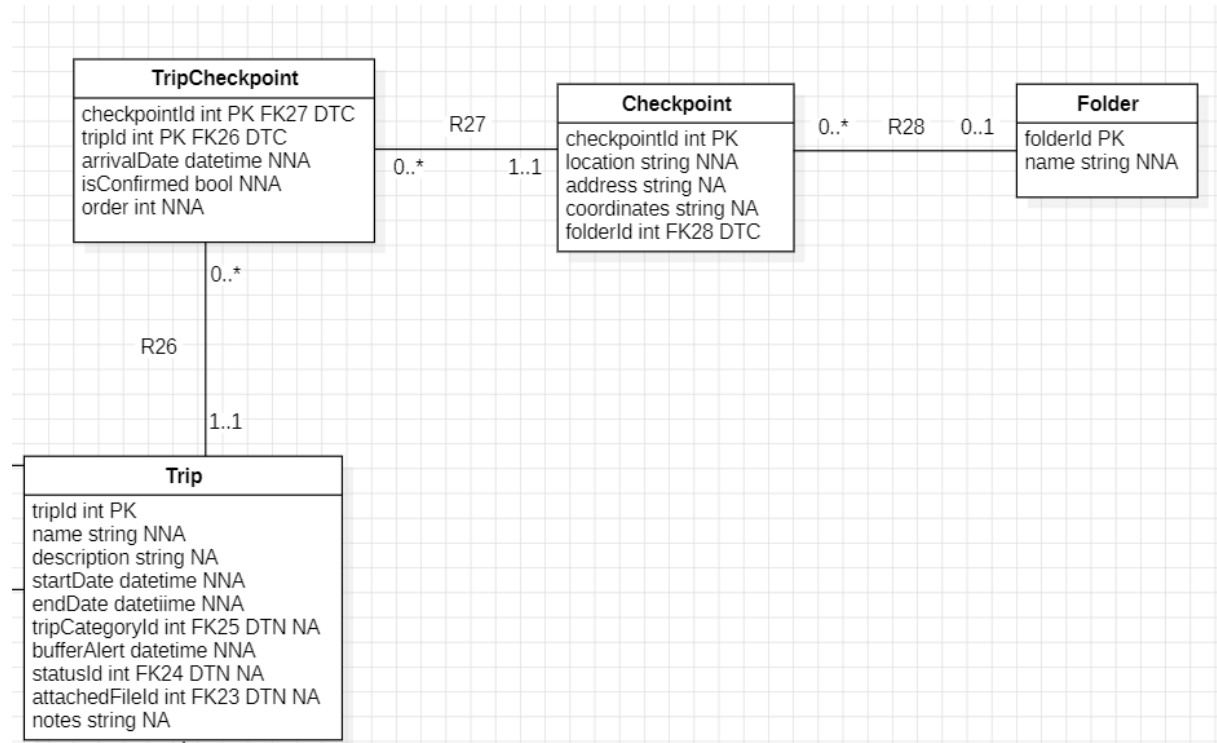


2.1.1.12. Manage checkpoints

Functionality: As a Family member, I can view checkpoints.

Preconditions: Actor is logged in as the System User

Data Model View:

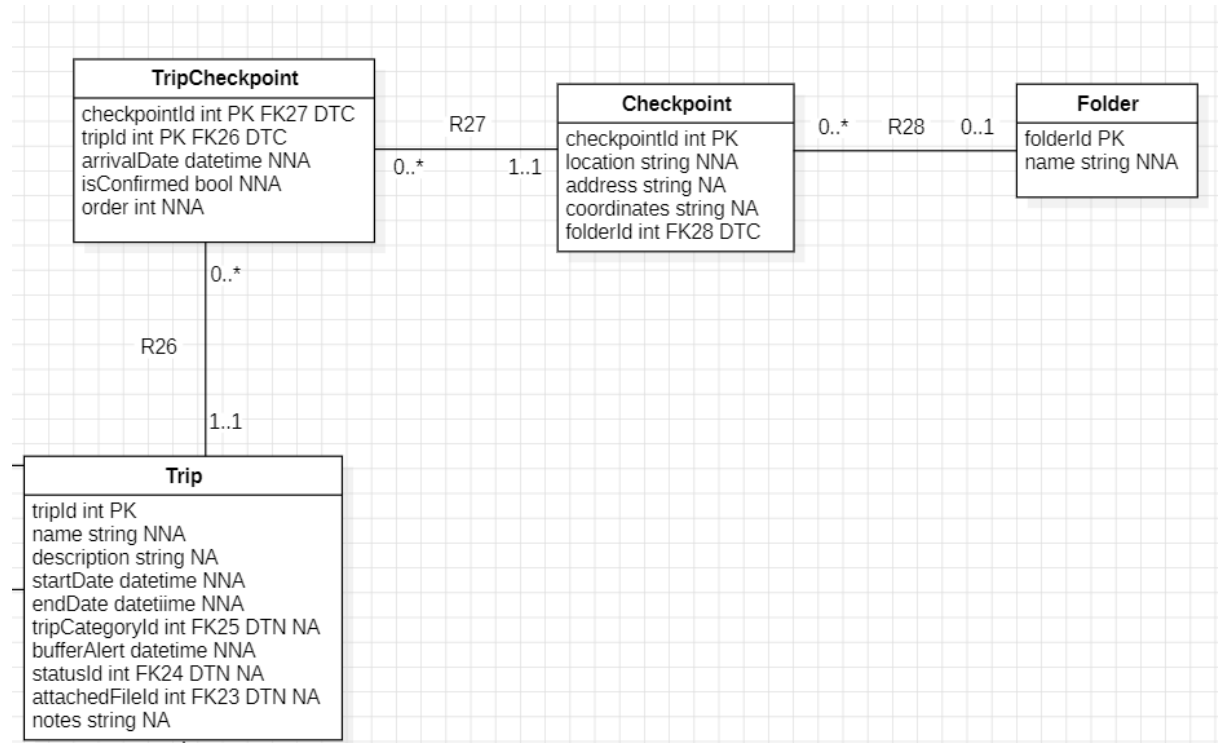


2.1.1.13. Confirm checkpoints

Functionality: As a Family member, I can confirm checkpoints.

Preconditions: Actor is logged in as the System User

Data Model View:

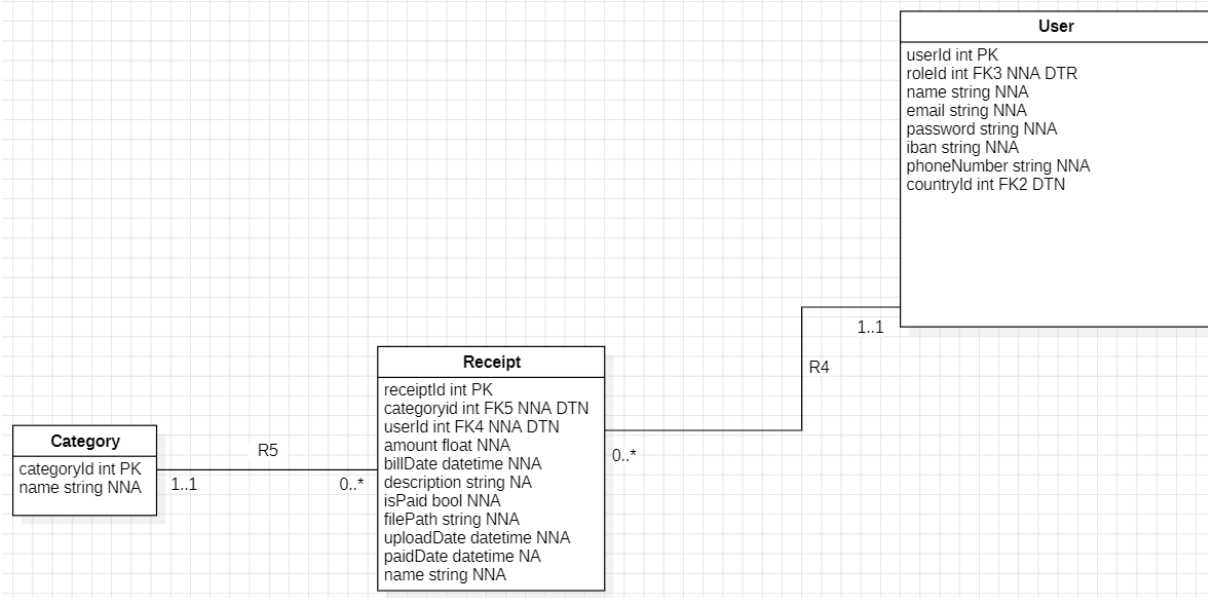


2.1.1.14. View invoices

Functionality: As The Andersons, I can view invoices.

Preconditions: Actor is logged in as the System User

Data Model View:

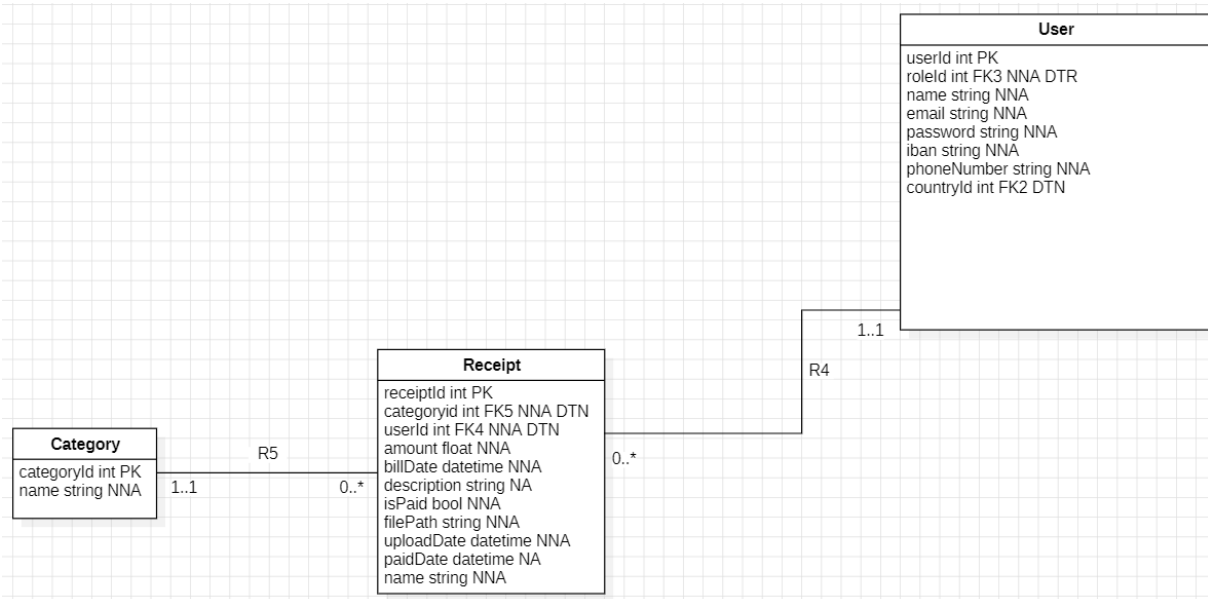


2.1.1.15. Approve invoices

Functionality: As The Andersons, I can approve invoices.

Preconditions: Actor is logged in as the System User

Data Model View:

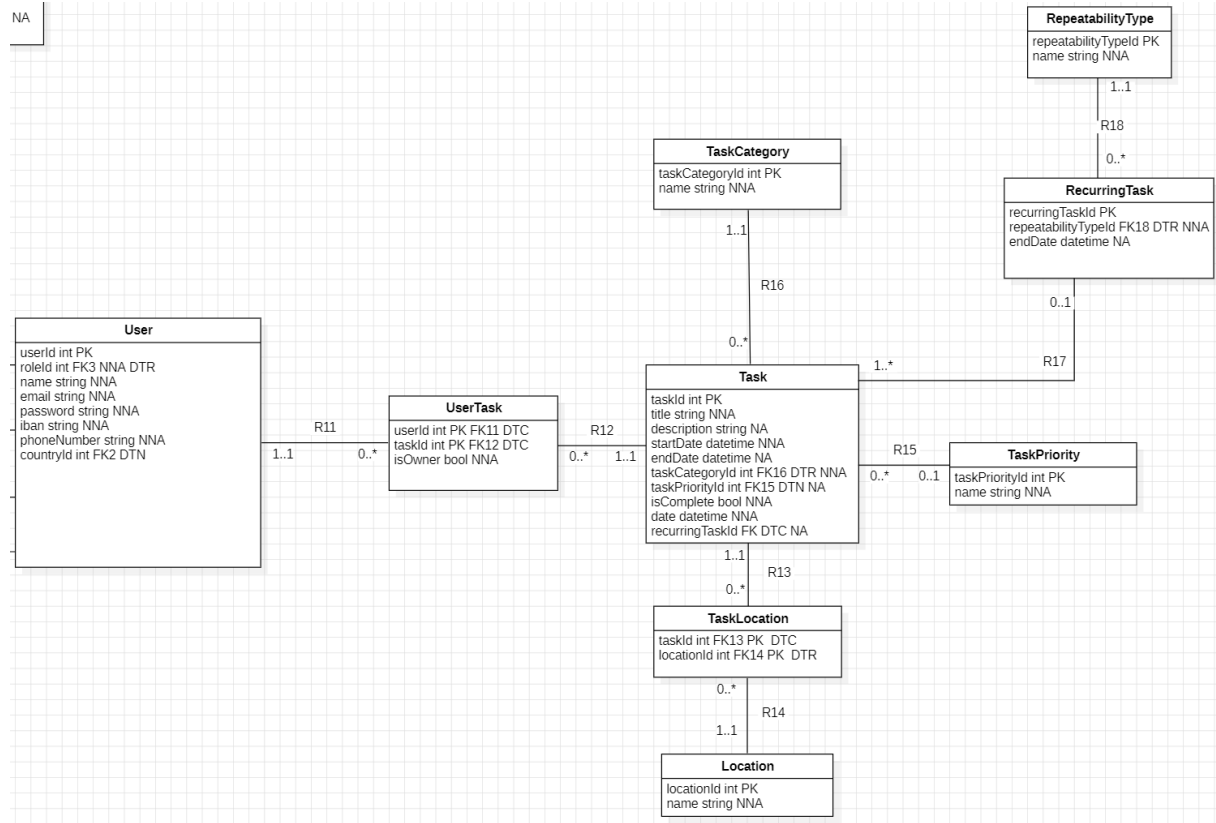


2.1.1.16. Manage Personal Tasks

Functionality: As a Staff Member, I can manage my personal tasks.

Precondition: Actor is logged in as a Staff Member

Data Model View:

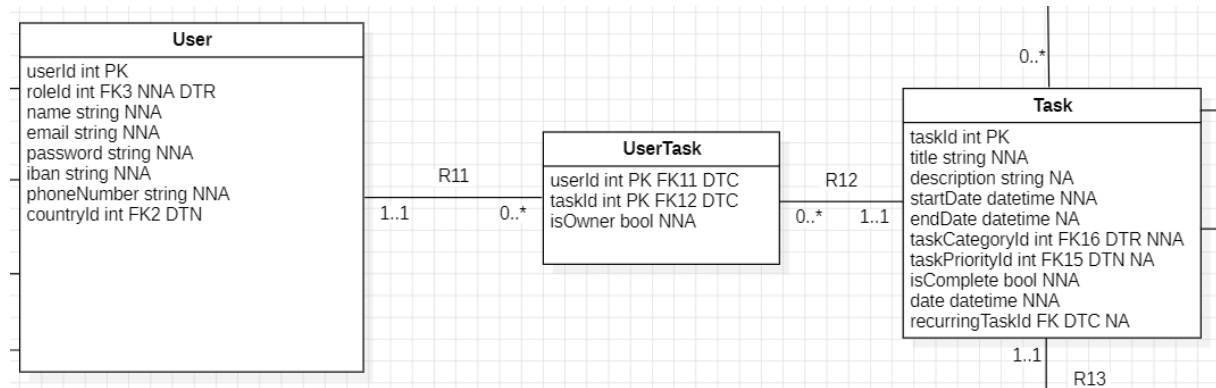


2.1.1.17. Request Collaboration

Functionality: As a Staff Member, I can manage request collaboration.

Preconditions: Actor is logged in as a Staff Member

Data Model View:

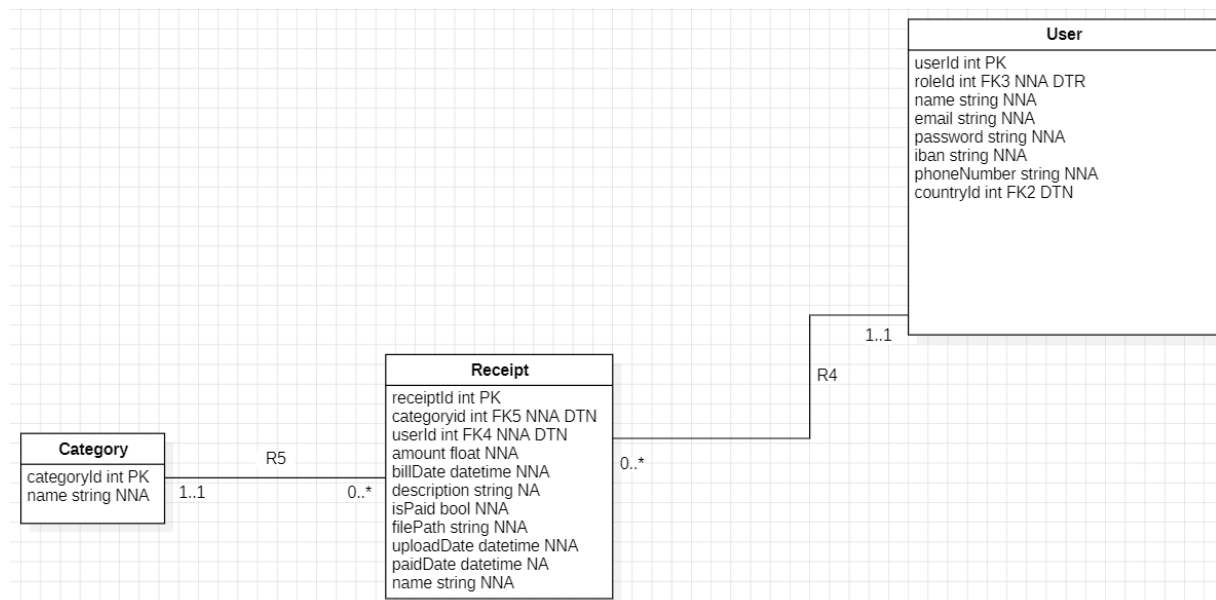


2.1.1.18. Manage Personal Invoices

Functionality: As a Staff Member, I can manage personal invoices.

Preconditions: Actor is logged in as a Staff Member

Data Model View:

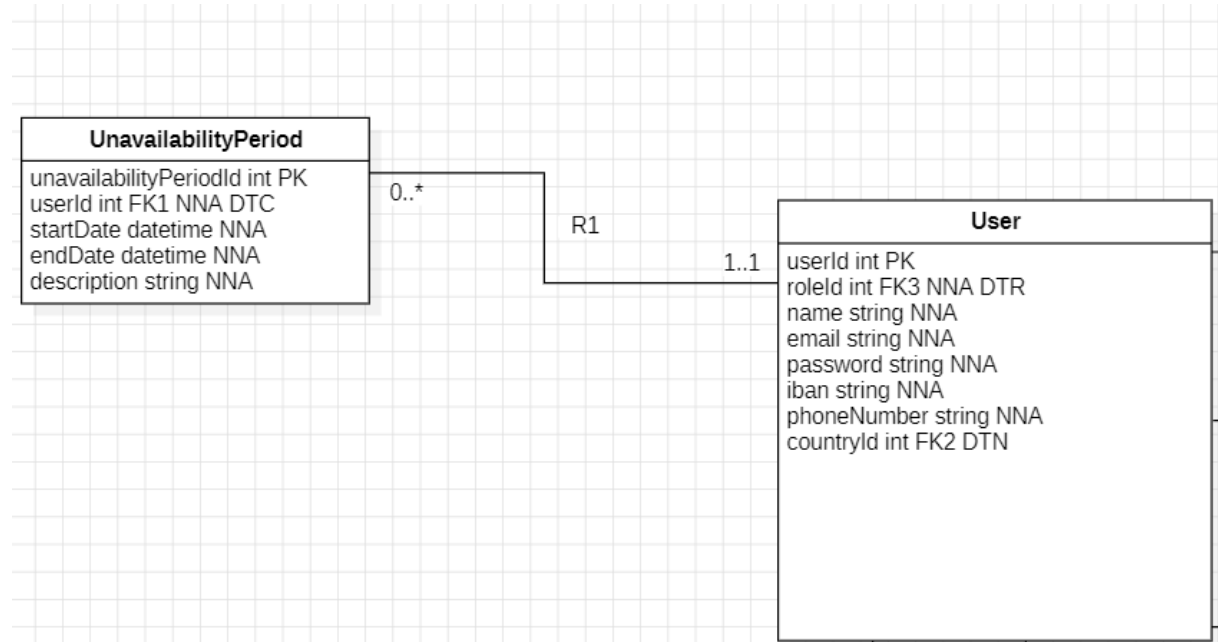


2.1.1.19. Manage Personal Availability

Functionality: As a Staff Member, I can manage personal availability

Preconditions: Actor is logged in as a Staff Member

Data Model View:

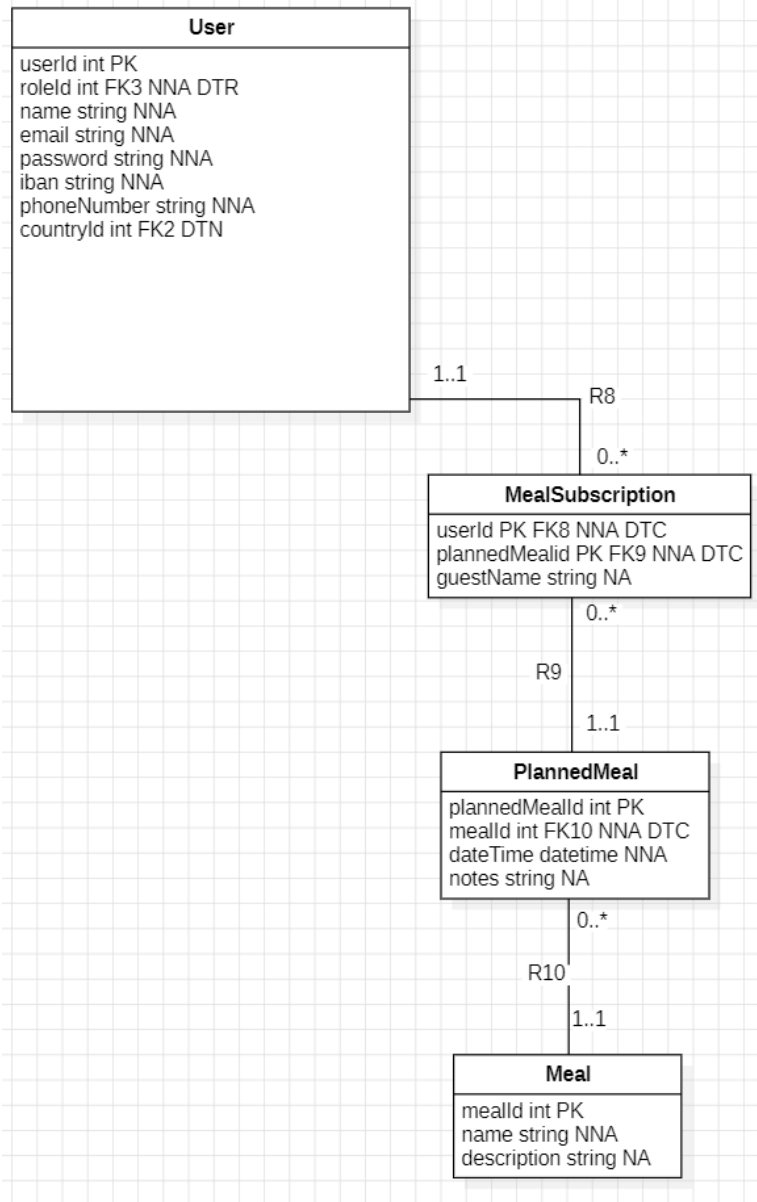


2.1.1.20. View Planned Meals

Functionality: As a Chef, I can View planned meals.

Preconditions: Actor is logged in as the Chef

Data Model View:

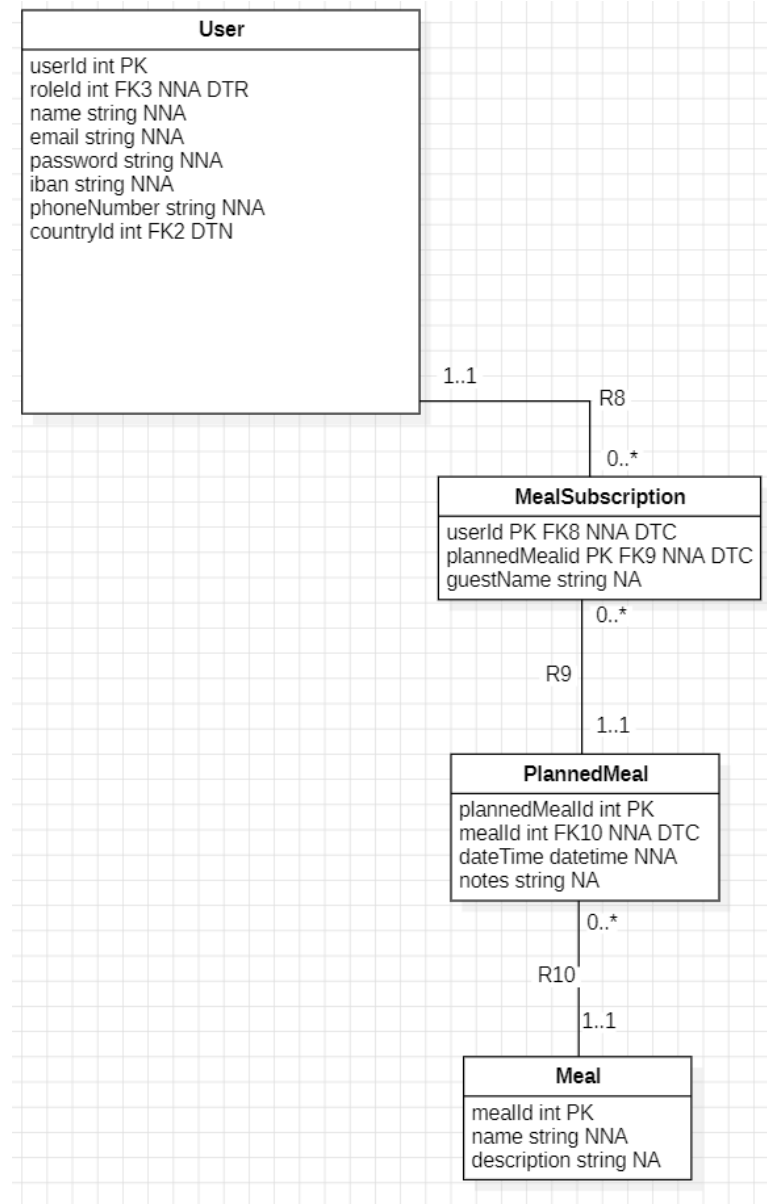


2.1.1.21. Confirm Planned Meal

Functionality: As a Chef, I can confirm planned meals.

Preconditions: Actor is logged in as the Chef

Data Model View:

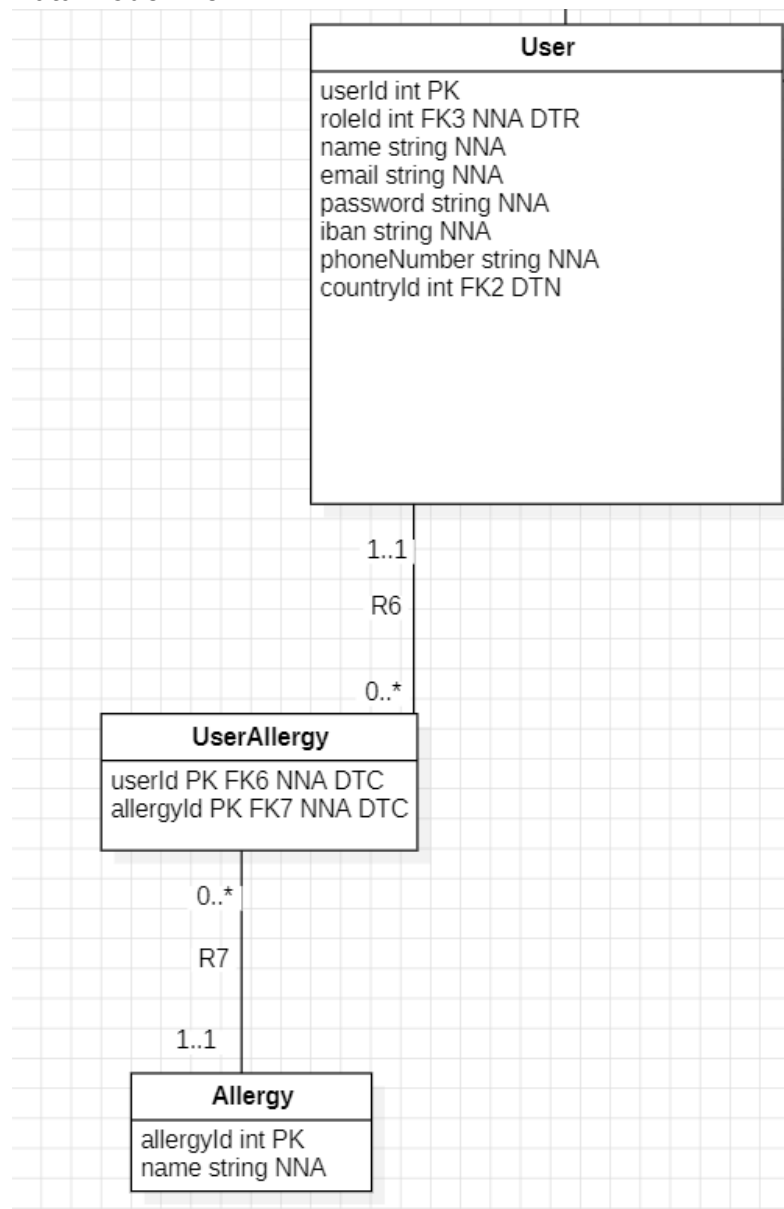


2.1.1.22. View all allergies

Functionality: As a Chef, I can view allergies of attendees.

Preconditions: Actor is logged in as the Chef

Data Model View:

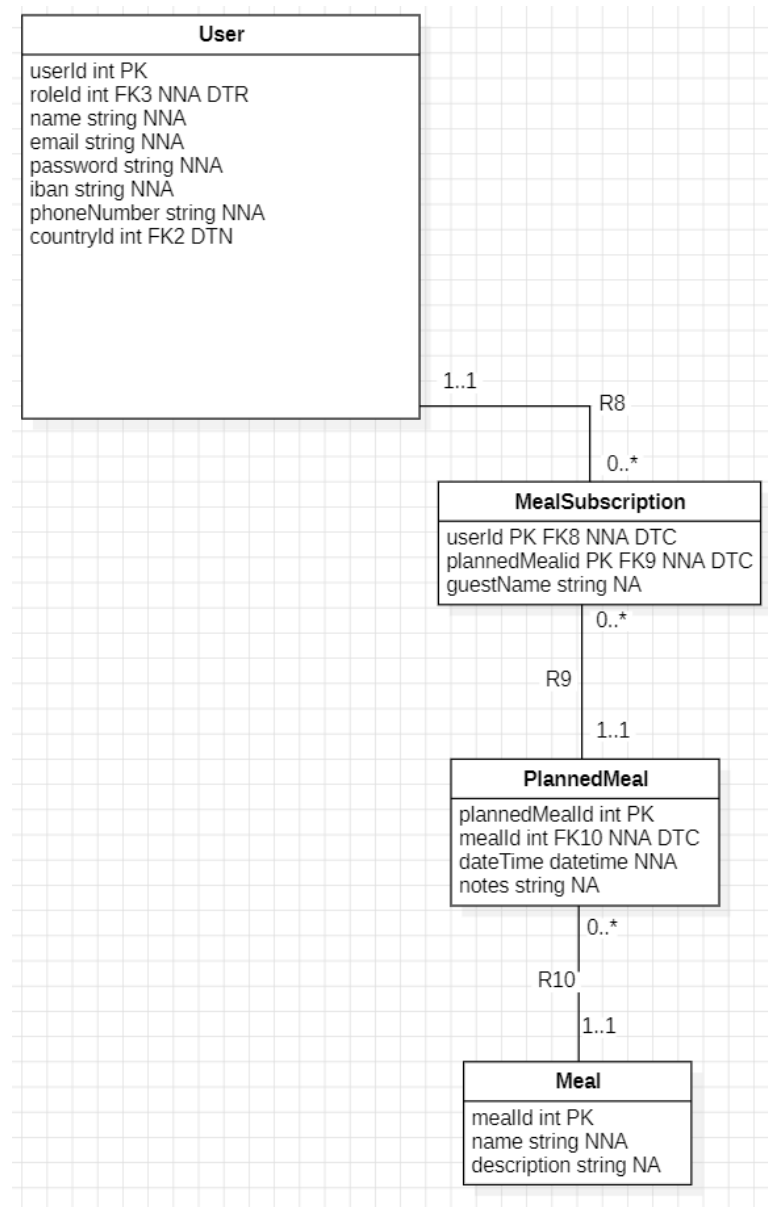


2.1.1.23. Manage Meal Plan

Functionality: As an Admin, I can manage meal.

Preconditions: Actor is logged in as the Admin

Data Model View:

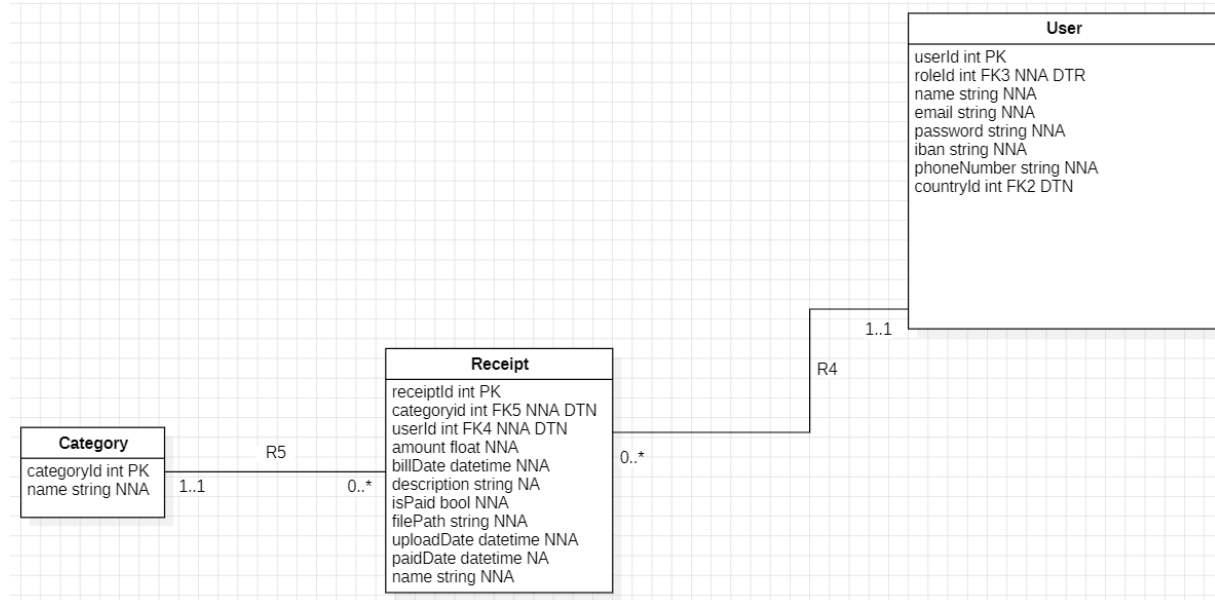


2.1.1.24. Manage Staff Invoices

Functionality: As an Admin, I can manage staff receipts.

Preconditions: Actor is logged in as the Admin

Data Model View:

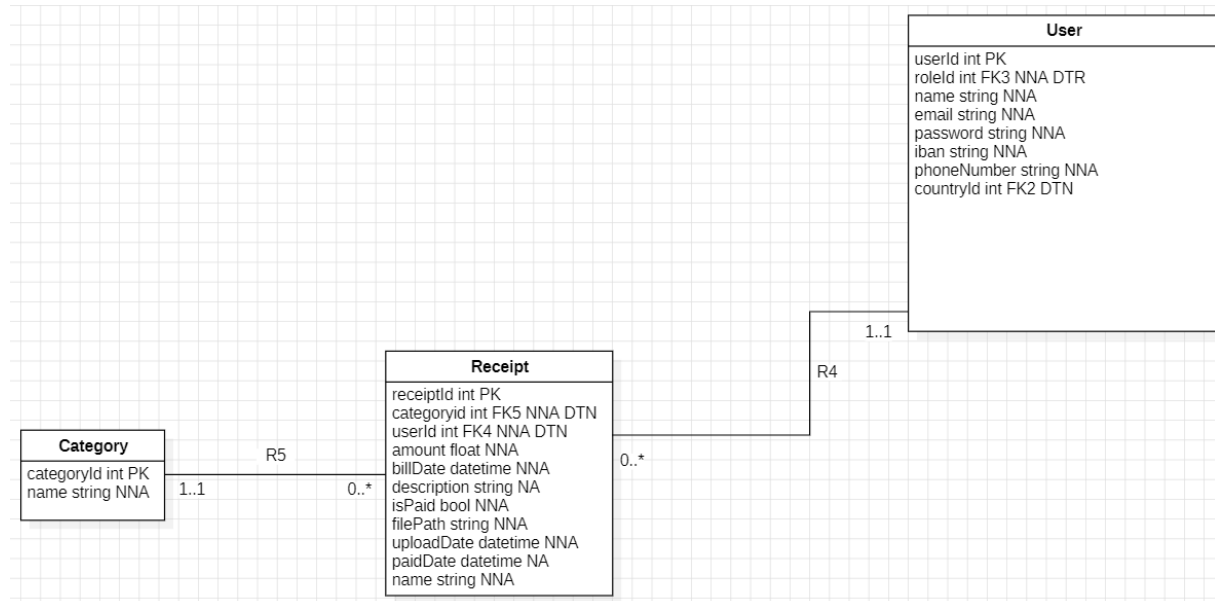


2.1.1.25. Print invoices

Functionality: As an Admin, I can print receipts.

Preconditions: Actor is logged in as the Admin

Data Model View:

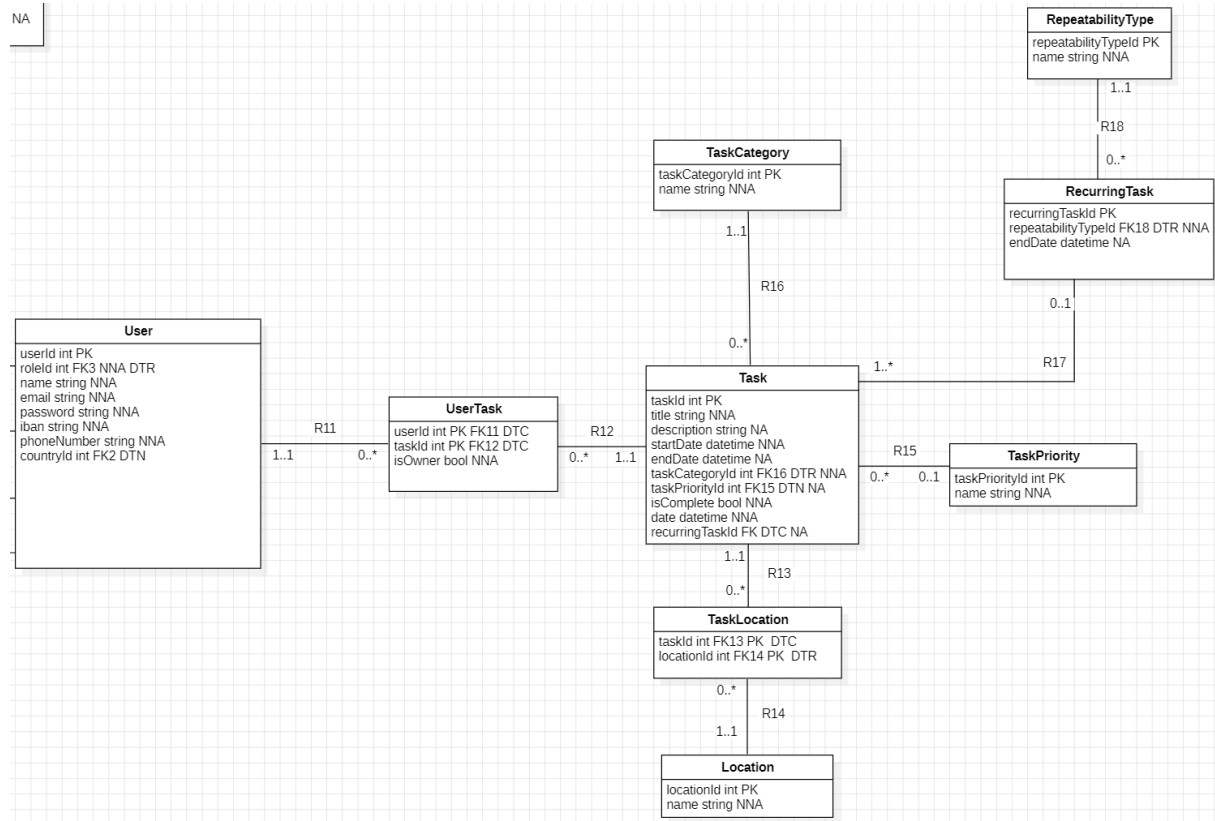


2.1.1.26. Manage Staff Tasks

Functionality: As an Admin, I can manage staff tasks

Preconditions: Actor is logged in as the Admin

Data Model View:

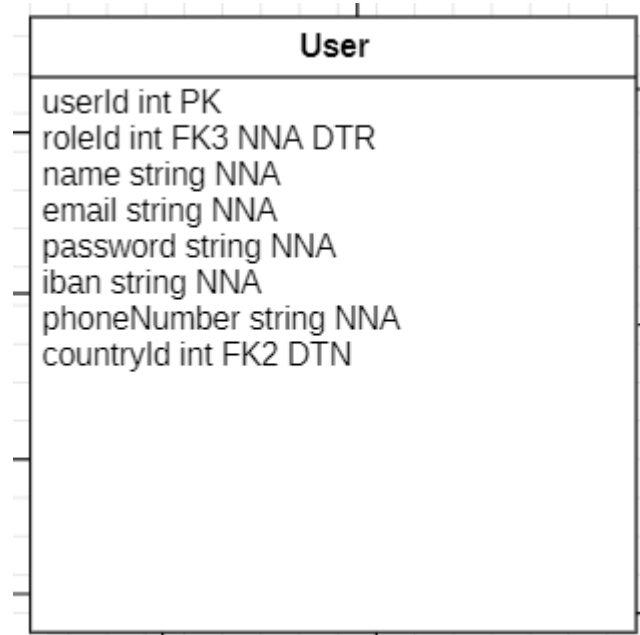


2.1.1.27. Manage Users

Functionality: As Admin, I can manage users

Preconditions: Actor is logged in as the Admin

Data Model View:

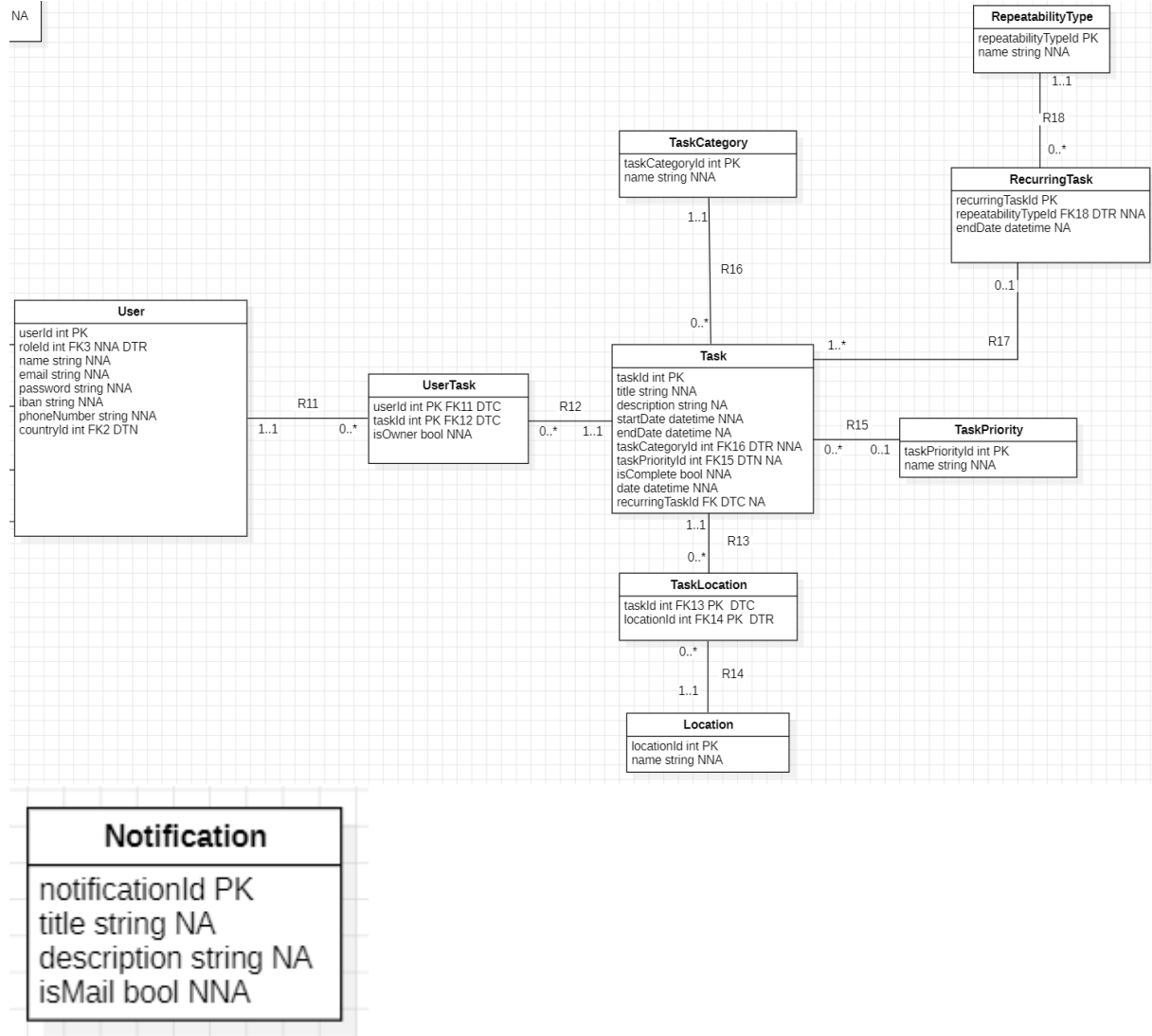


2.1.1.28. Send task reminder

Functionality: As Time, I can send a task reminder

Preconditions: There is an at least one task scheduled

Data Model View:



2.2. Non-functional requirements

2.2.1. Usability & Accessibility

- The system should be usable by elderly users with limited digital experience.
- The interface should use large fonts and sufficient spacing to improve readability.
- The color blue should be used as the primary interface color, while red should be reserved for alerts only.
- All essential functionality should be accessible within a minimal number of interactions.

2.2.2. Performance

- The system should provide fast feedback when users add or modify schedules, dinners, or availability.
- The application should remain responsive during normal household usage.

2.2.3. Reliability

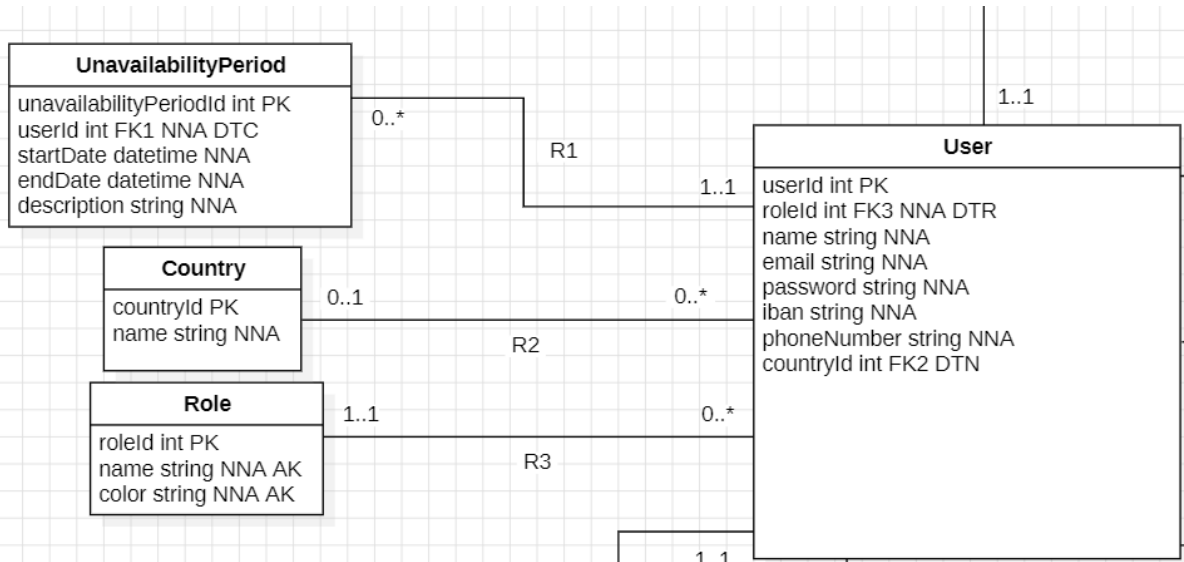
- The system should prevent loss of scheduled data such as worker appointments and invoices.
- Changes made by users should be saved reliably.

2.2.4. Security & Privacy

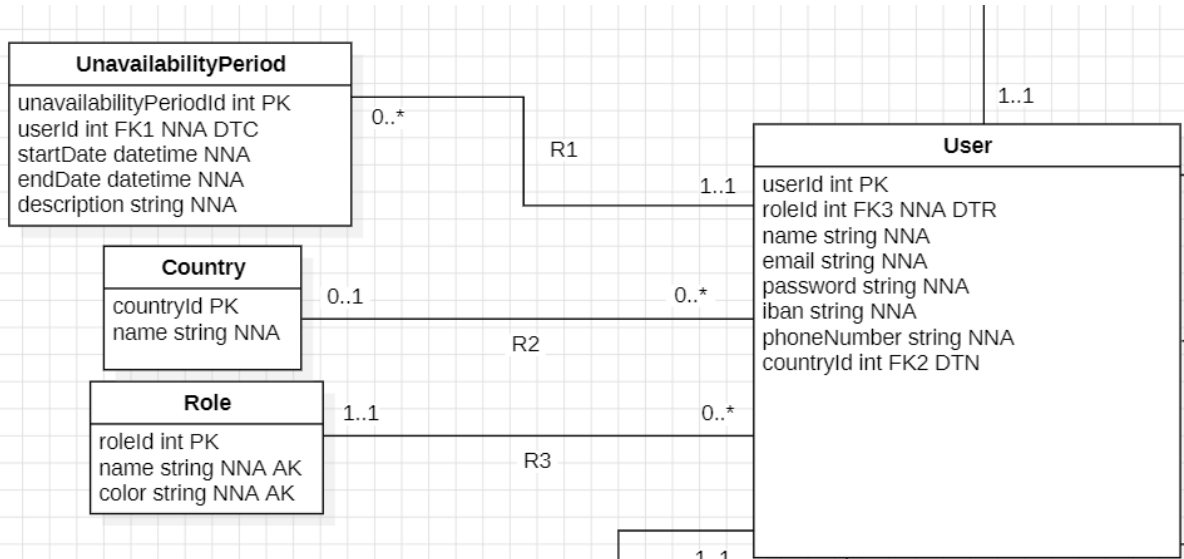
- Personal and financial data should be protected against unauthorized access.
- The system should comply with applicable data protection regulations (e.g. GDPR).

3. Data Model

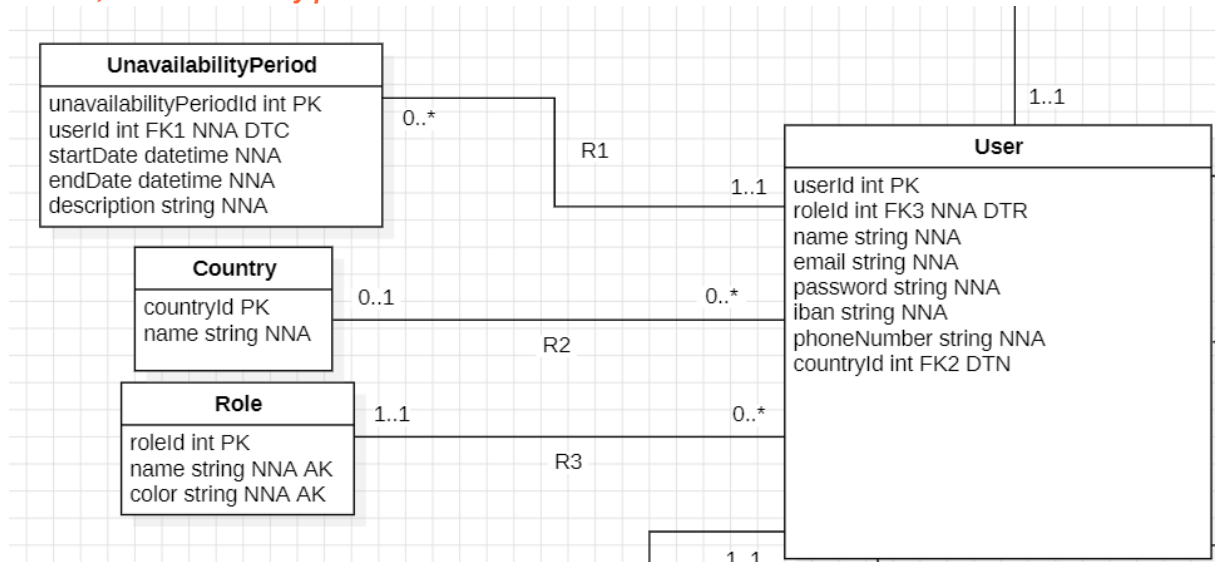
***R1: A user belongs to one country, and a country can have many users.
countryId FK2 DTN: If a country is deleted, all users from that country will no longer have a country assigned. The users will continue to exist, but without a country.***



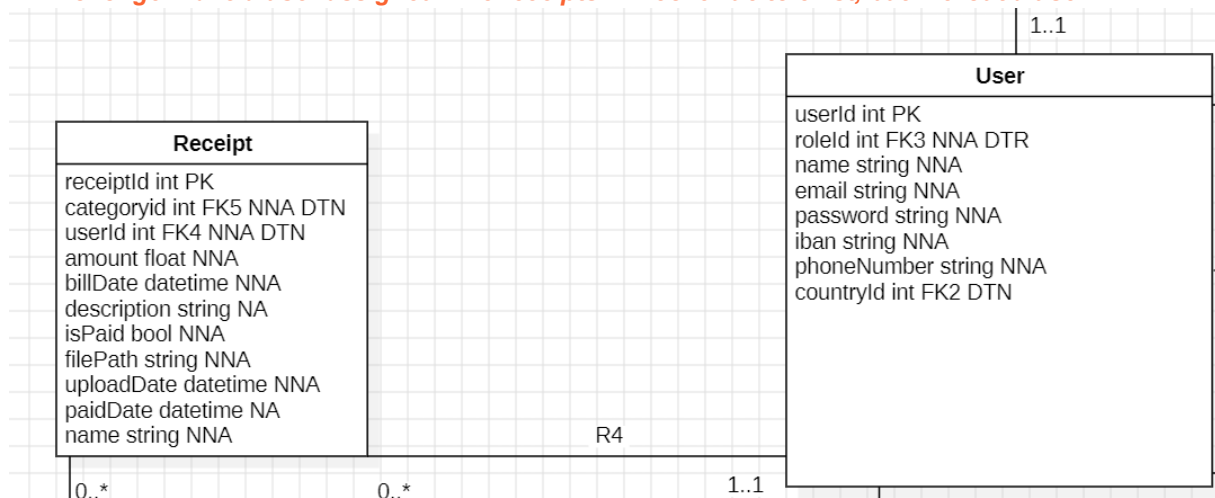
R2 : A user has one role, and a role can belong to many users.
roleId FK3 NNA DTR: If a user is created, the role must be known. A role cannot be deleted as long as there are users with that role.



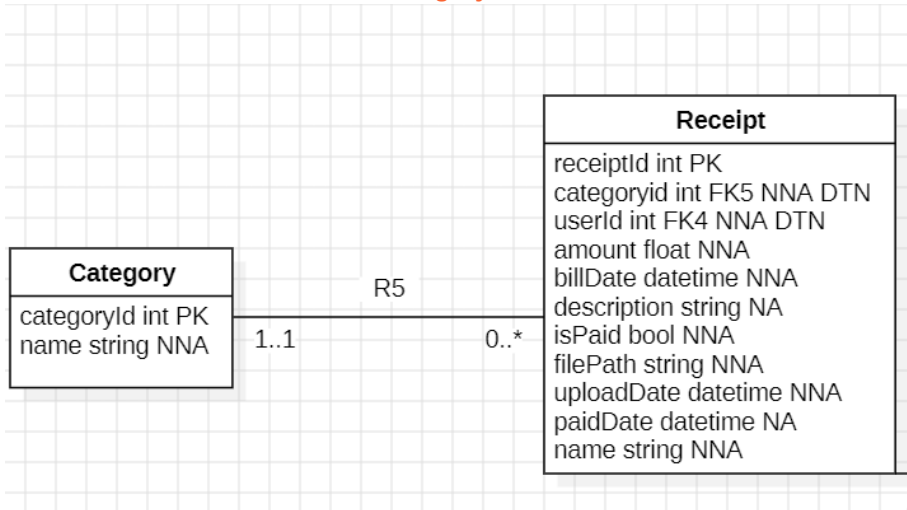
R3: A user can have many unavailability periods, and an unavailability period belongs to one user. *userId FK1 NNA DTC: If an unavailability period is created, the user must be known. If a user is deleted, all unavailability periods of that user are deleted as well.*



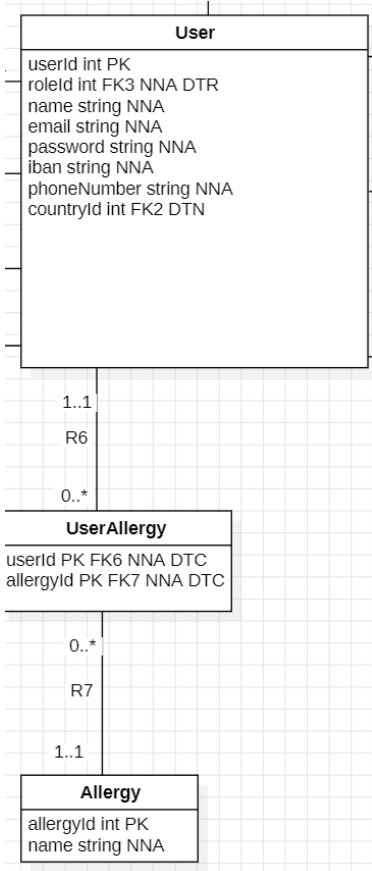
R4: A user can have many receipts, and a receipt belongs to one user. *userId FK4 NNA DTN: If a receipt is created, the user must be known. If a user is deleted, all receipts will no longer have a user assigned. The receipts will continue to exist, but without a user.*



R5: A receipt belongs to one category, and a category can have many receipts. categoryId FK5 NNA DTN: If a receipt is created, the category must be known. If a category is deleted, all receipts in that category will no longer have a category assigned. The receipts will continue to exist, but without a category.



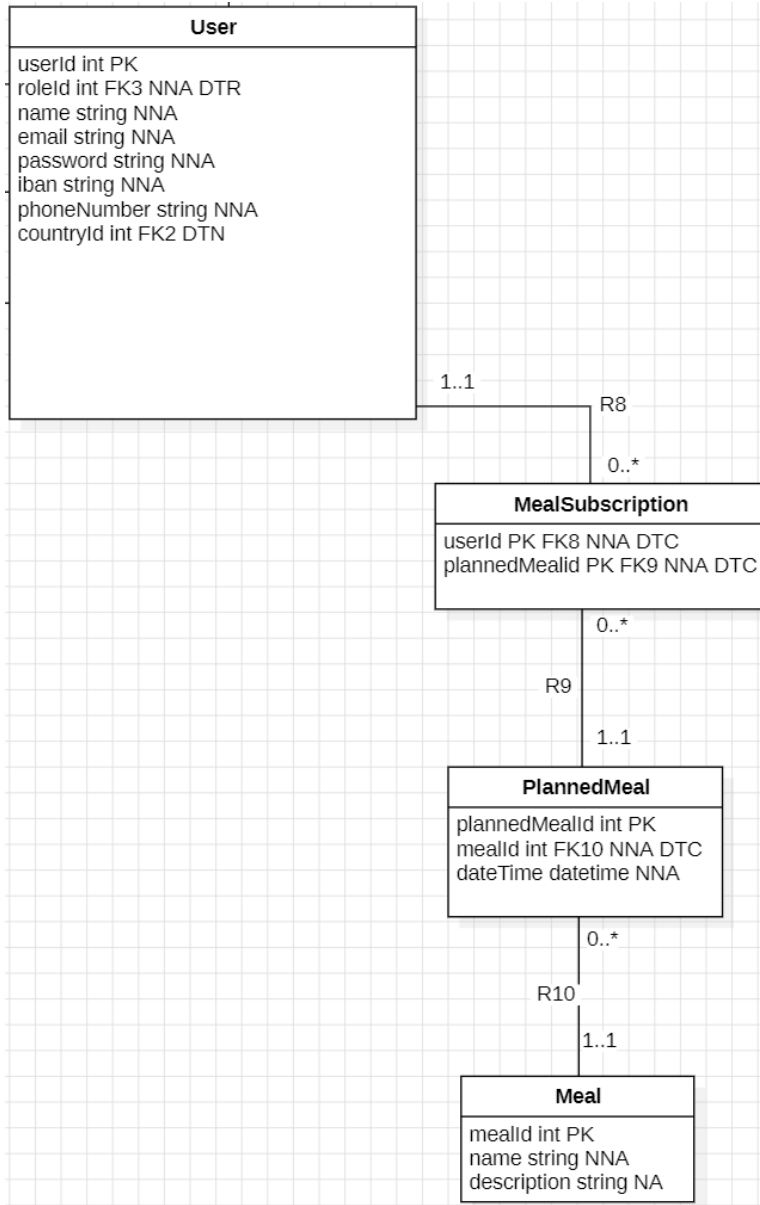
R6 + R7: A user can have many allergies, and an allergy can belong to many users. userId FK6 NNA DTC: If a user-allergy association is created, the user must be known. If a user is deleted, all allergy associations for that user are deleted as well. allergyId FK7 NNA DTC: If a user-allergy association is created, the allergy must be known. If an allergy is deleted, all user associations for that allergy are deleted as well.



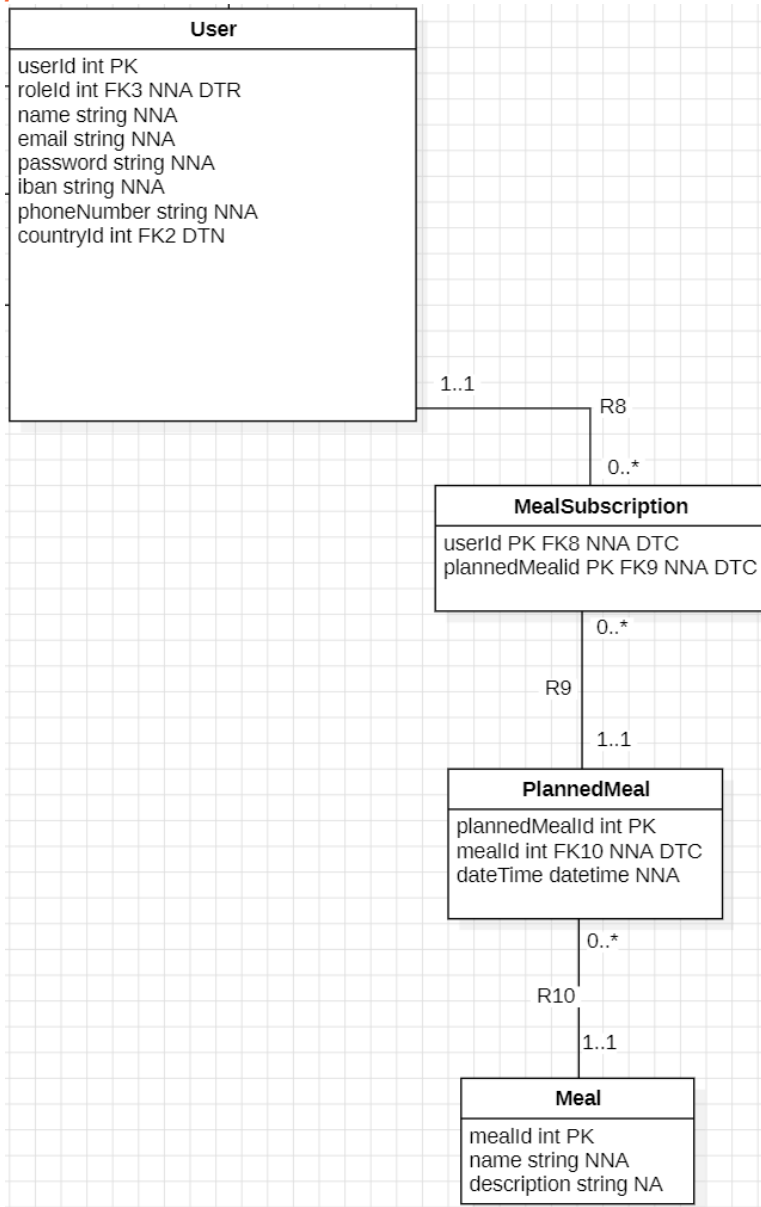
R8 + R9: A user can subscribe to many planned meals, and a planned meal can have many user subscriptions.

userId FK8 NNA DTC: If a meal subscription is created, the user must be known. If a user is deleted, all meal subscriptions for that user are deleted as well.

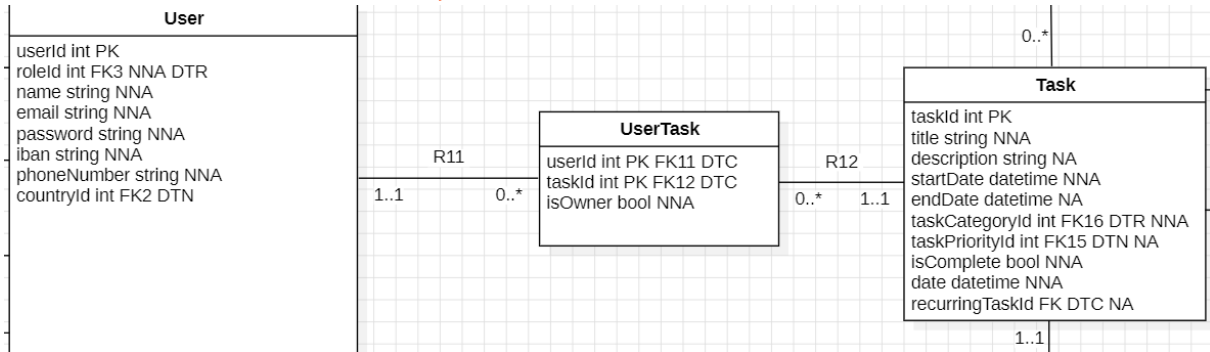
plannedMealId FK9 NNA DTC: If a meal subscription is created, the planned meal must be known. If a planned meal is deleted, all subscriptions for that meal are deleted as well.



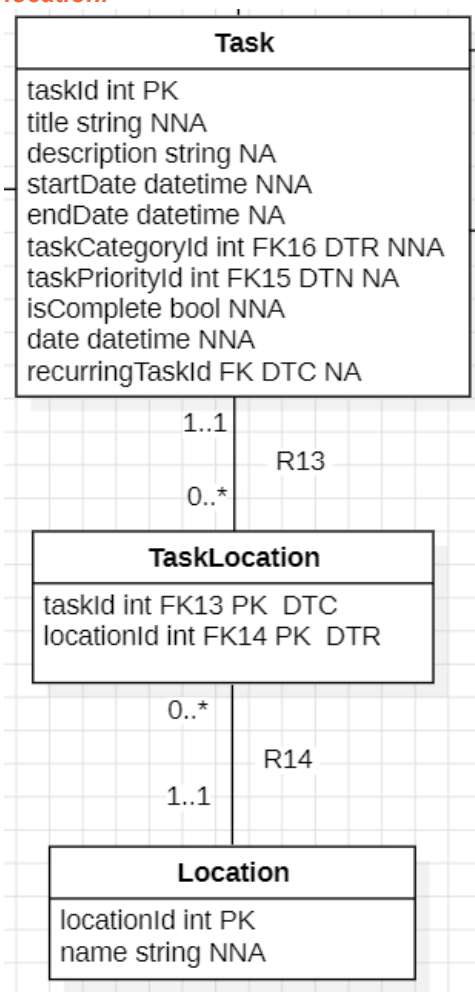
R10: A planned meal references one meal, and a meal can be referenced by many planned meals. mealId FK10 NNA DTC: If a planned meal is created, the meal must be known. If a meal is deleted, all planned meals for that meal are deleted as well.



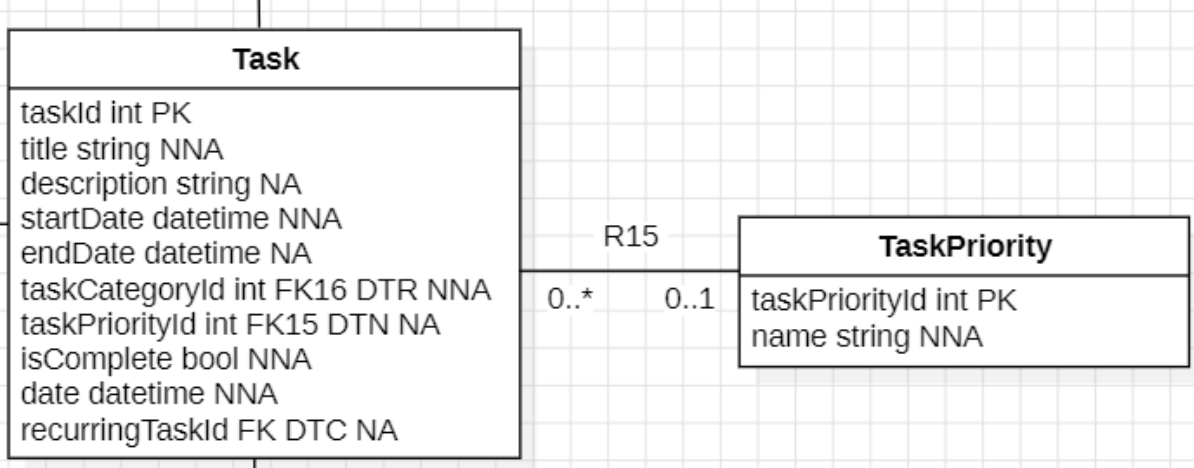
R11 + R12: A user can have many tasks, and a task can belong to many users.
userId FK11 DTC: If a user is deleted, all task associations for that user are deleted as well.
taskId FK12 DTC: If a task is deleted, all user associations for that task are deleted as well.



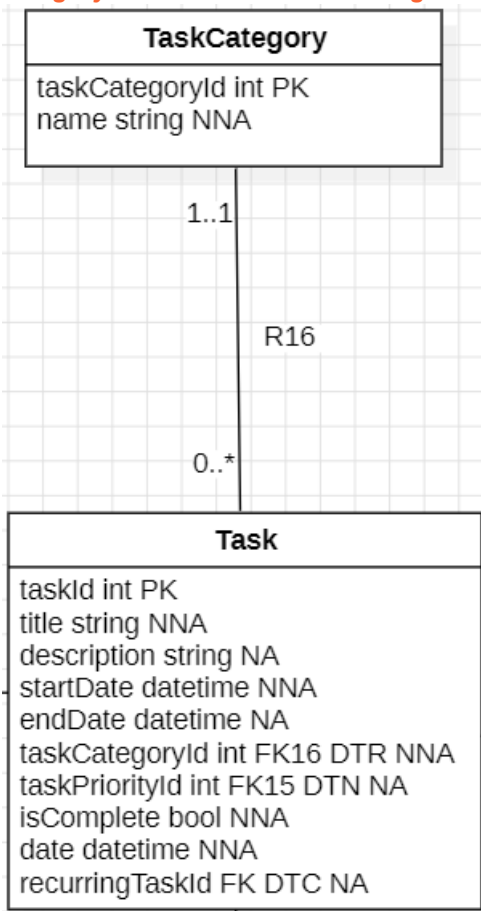
R13 + R14: A task can have many locations, and a location can belong to many tasks.
taskId FK13 DTC: If a task is deleted, all location associations for that task are deleted as well.
locationId FK14 DTR: A location cannot be deleted as long as there are tasks associated with that location.



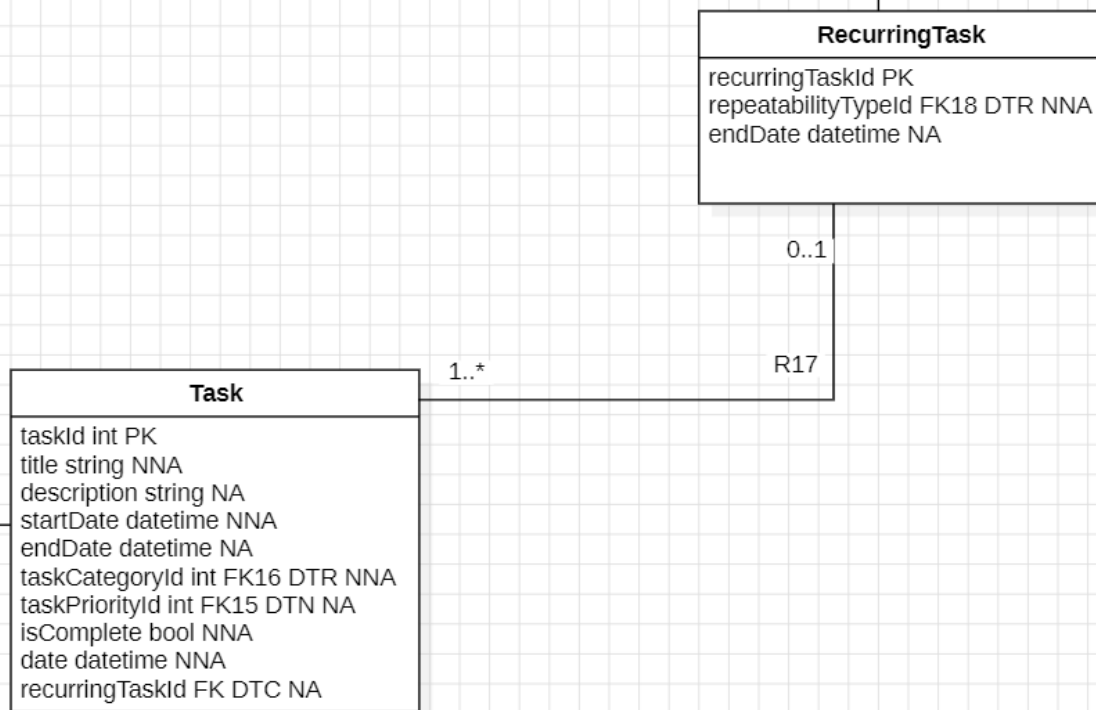
R15: A task has one task priority, and a task priority can belong to many tasks.
taskPriorityId FK15 DTN NA: If a task is created, the task priority does not need to be known. If a task priority is deleted, all tasks with that priority will no longer have a priority assigned. The tasks will continue to exist, but without a priority.



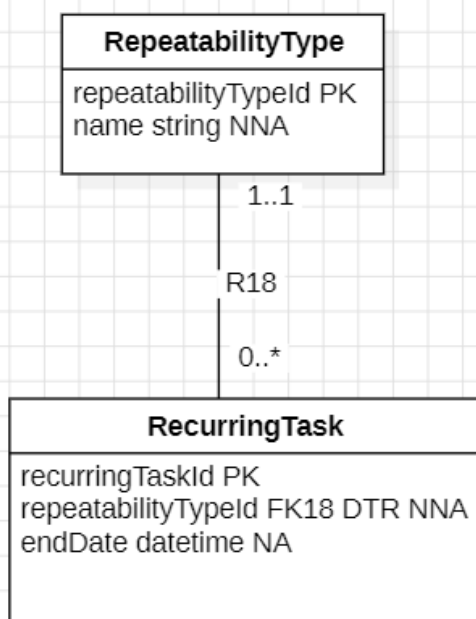
R16: A task belongs to one task category, and a task category can have many tasks.
taskCategoryId FK16 DTR NNA: If a task is created, the task category must be known. A task category cannot be deleted as long as there are tasks in that category.



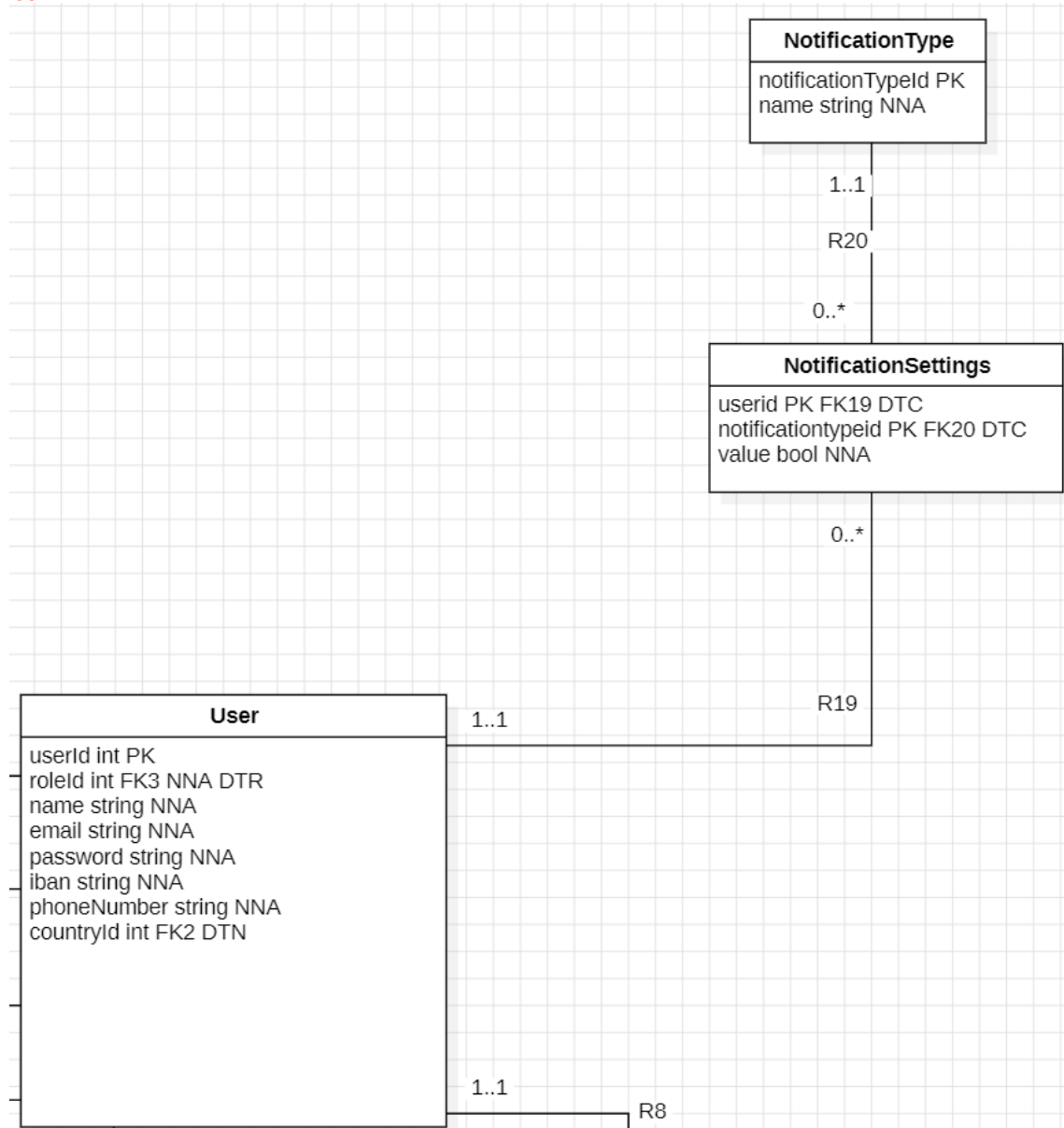
R17: A task can have one recurring task, and a recurring task belongs to one task.
recurringTaskId FK DTC NA: If a task is created, the recurring task does not need to be known. If a recurring task is deleted, all tasks linked to that recurring task are deleted as well.



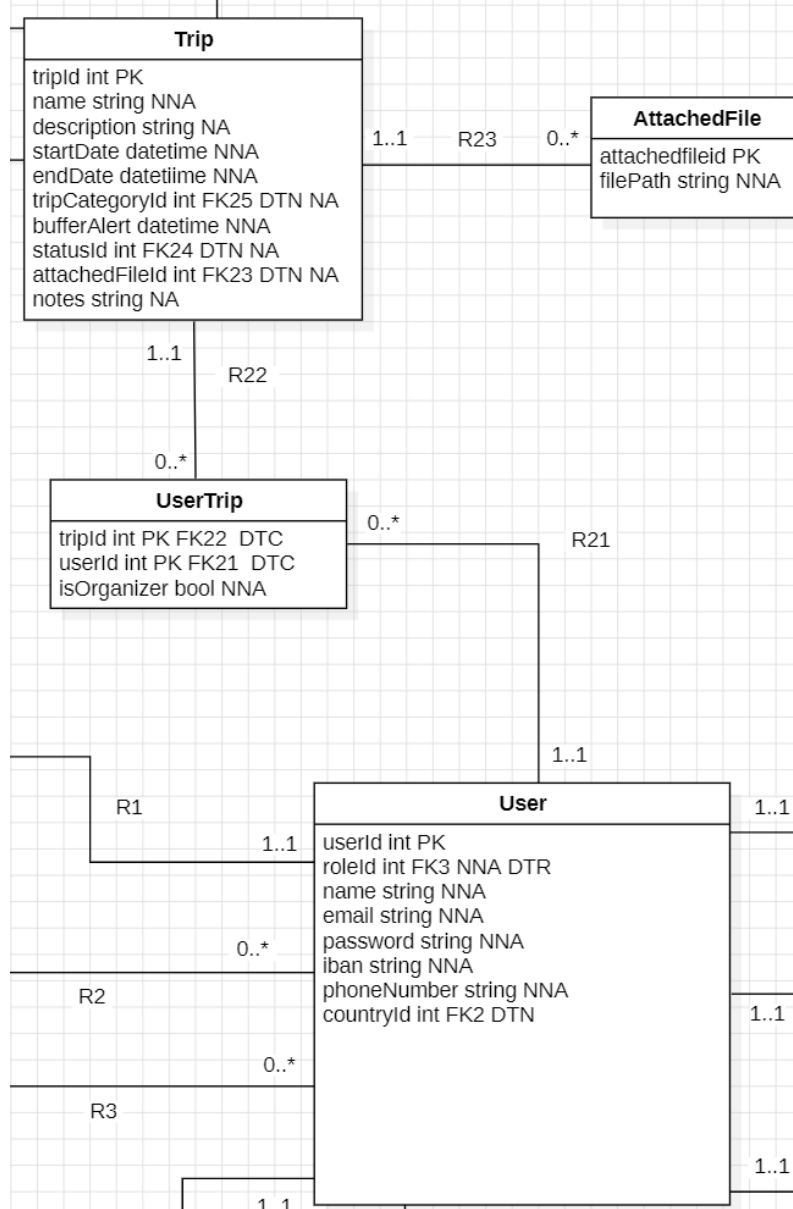
R18: A recurring task has one repeatability type, and a repeatability type can belong to many recurring tasks.
repeatabilityTypeId FK18 DTR NNA: If a recurring task is created, the repeatability type must be known. A repeatability type cannot be deleted as long as there are recurring tasks using that type.



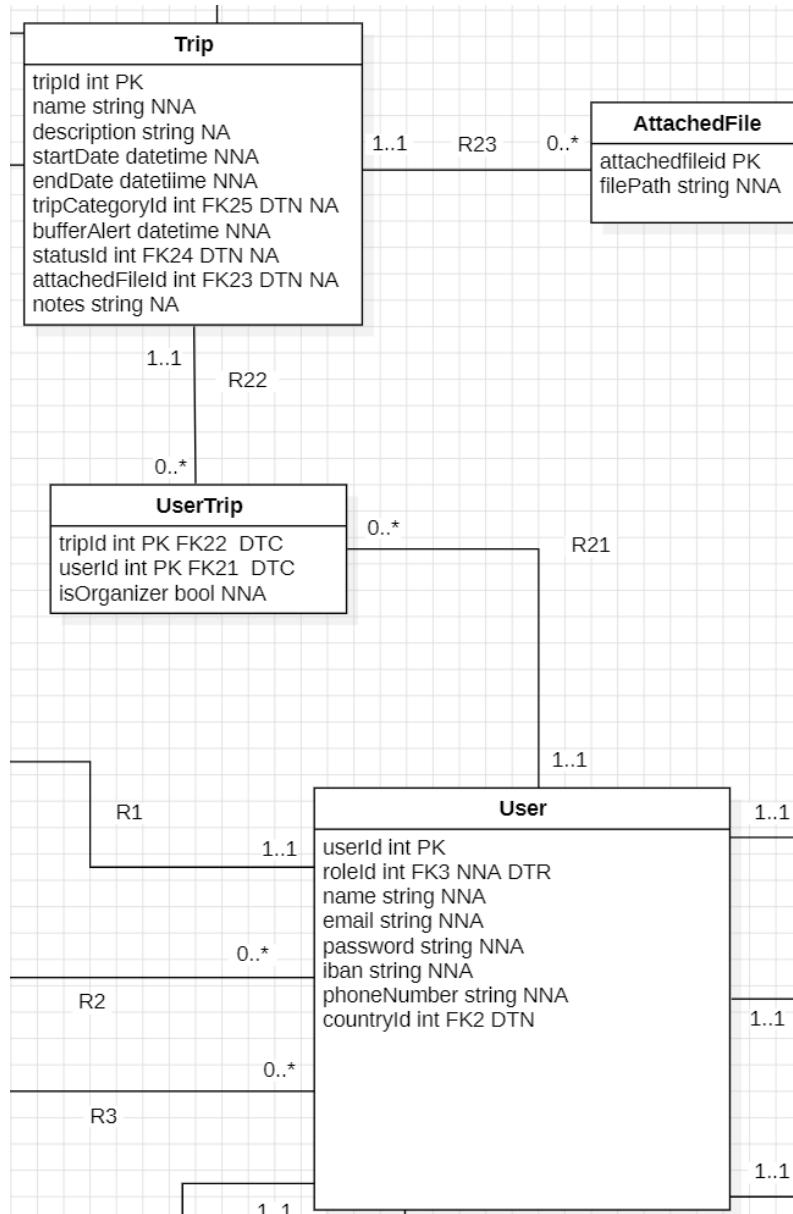
R19 + R20: A user can have many notification type settings, and a notification type can be configured by many users.
userId FK19 DTC: If a user is deleted, all notification settings for that user are deleted as well.
notificationTypeId FK20 DTC: If a notification type is deleted, all user settings for that notification type are deleted as well.



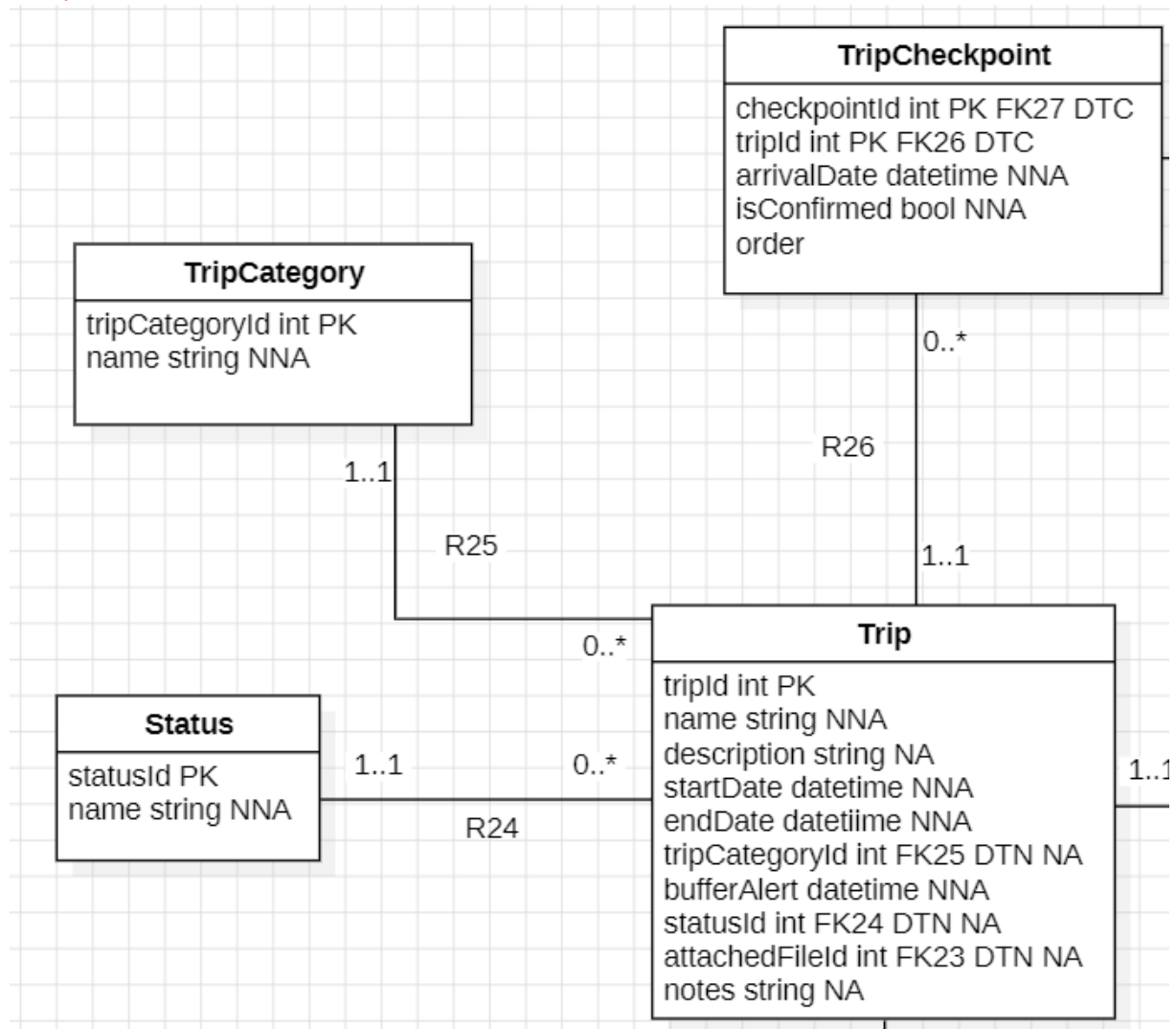
R21 + R22: A user can participate in many trips, and a trip can have many users.
userId FK21 DTC: If a user is deleted, all trip participations for that user are deleted as well.
tripId FK22 DTC: If a trip is deleted, all user participations for that trip are deleted as well.



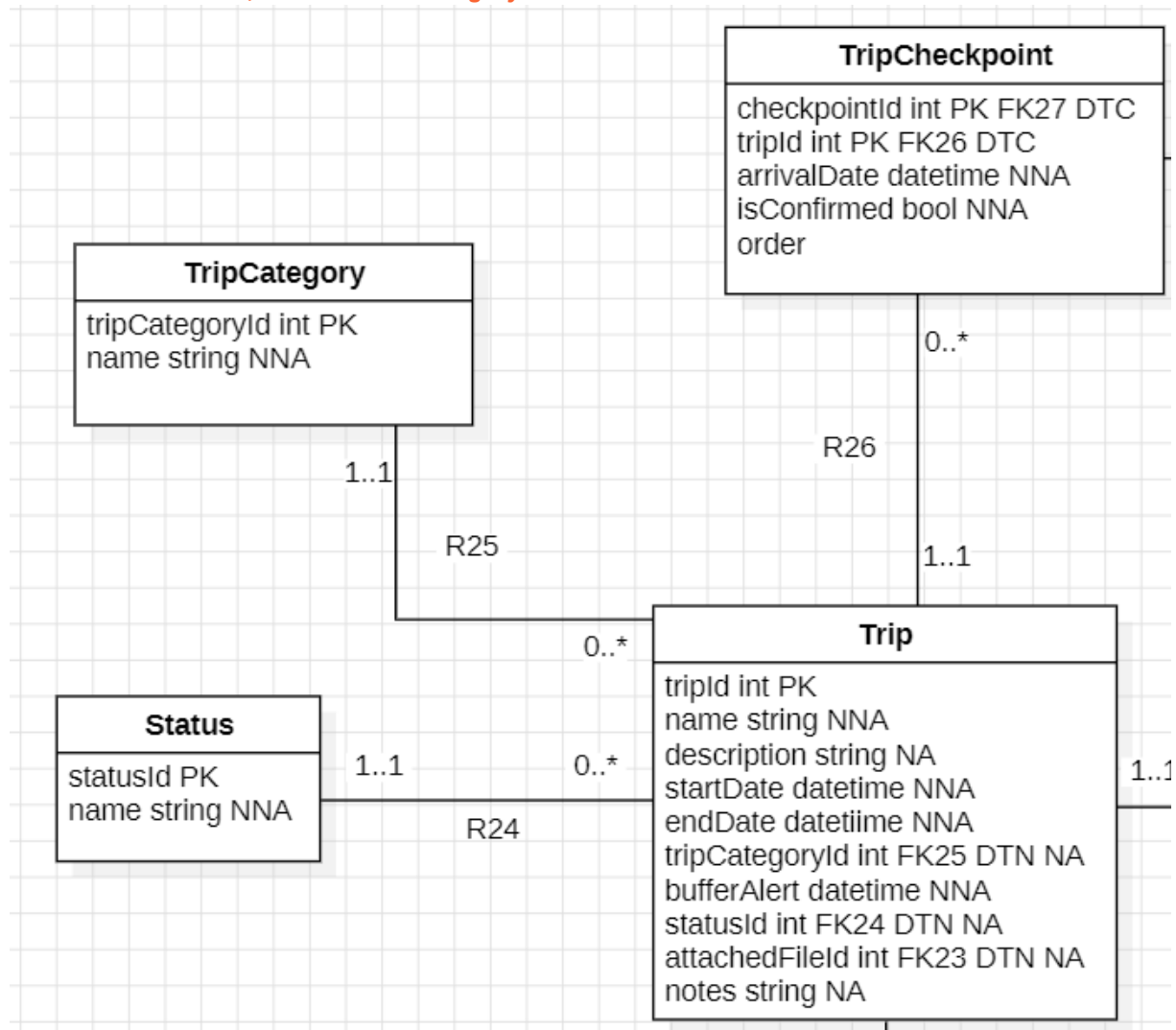
R23: A trip can have many attached files, and an attached file belongs to one trip.
attachedFileId FK23 DTN NA: If a trip is created, the attached file does not need to be known. If an attached file is deleted, the trip will no longer have that file attached. The trip will continue to exist, but without the attached file.



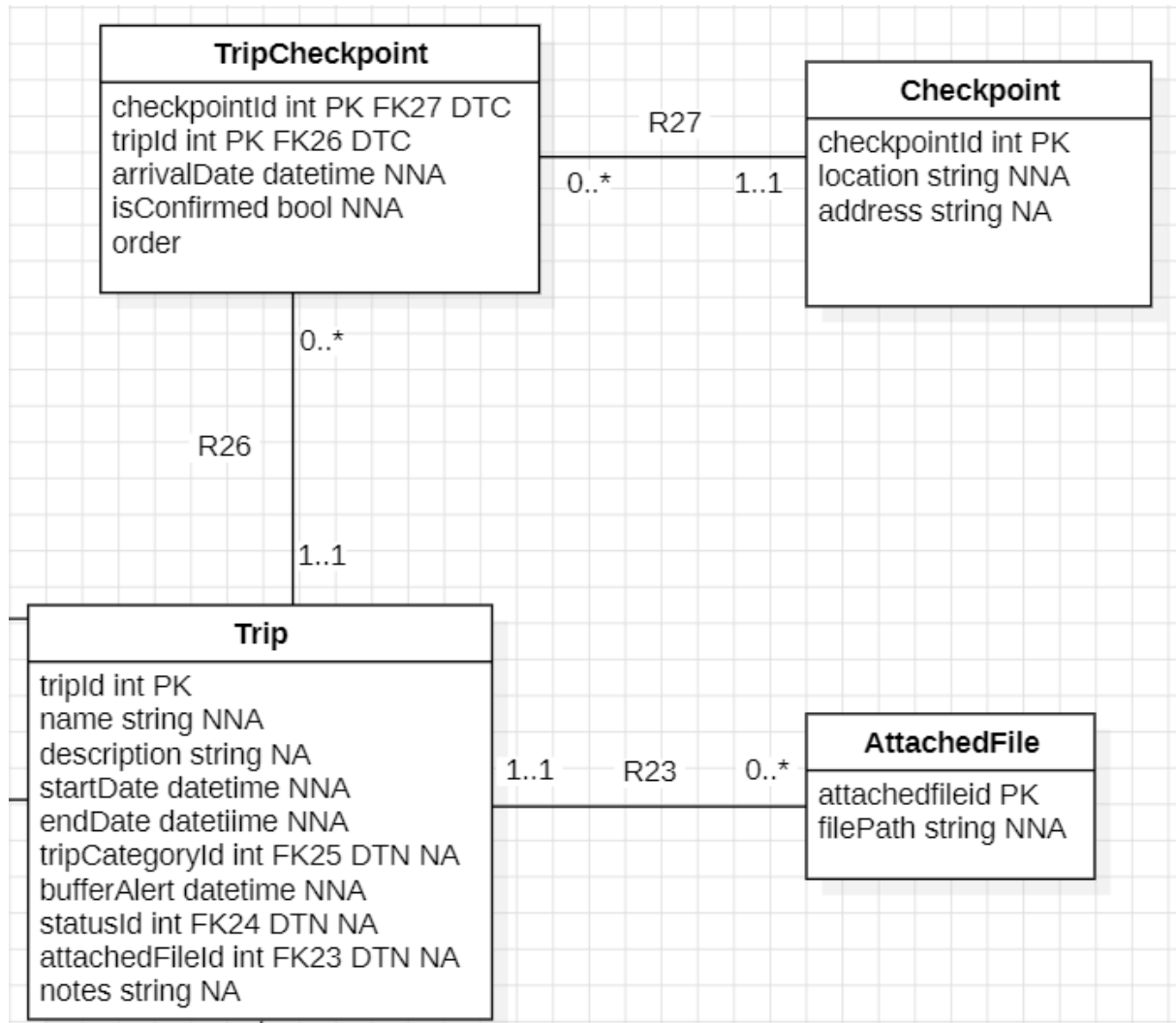
R24: A trip has one status, and a status can belong to many trips.
statusId FK24 DTN NA: If a trip is created, the status does not need to be known. If a status is deleted, all trips with that status will no longer have a status assigned. The trips will continue to exist, but without a status.



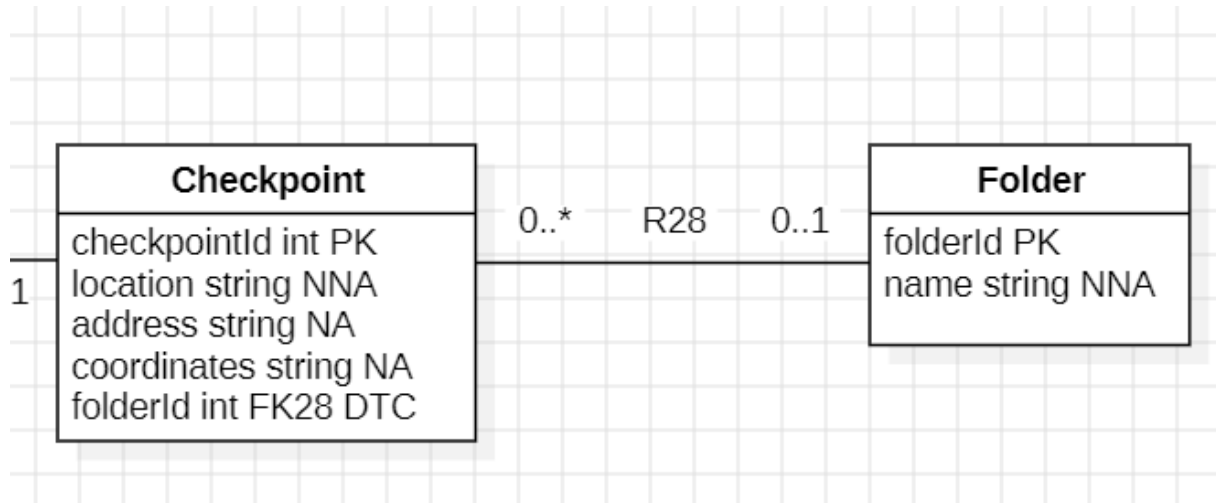
*R25: A trip belongs to one trip category, and a trip category can have many trips.
 tripCategoryId FK25 DTN NA: If a trip is created, the trip category does not need to be known. If a trip category is deleted, all trips in that category will no longer have a category assigned. The trips will continue to exist, but without a category.*



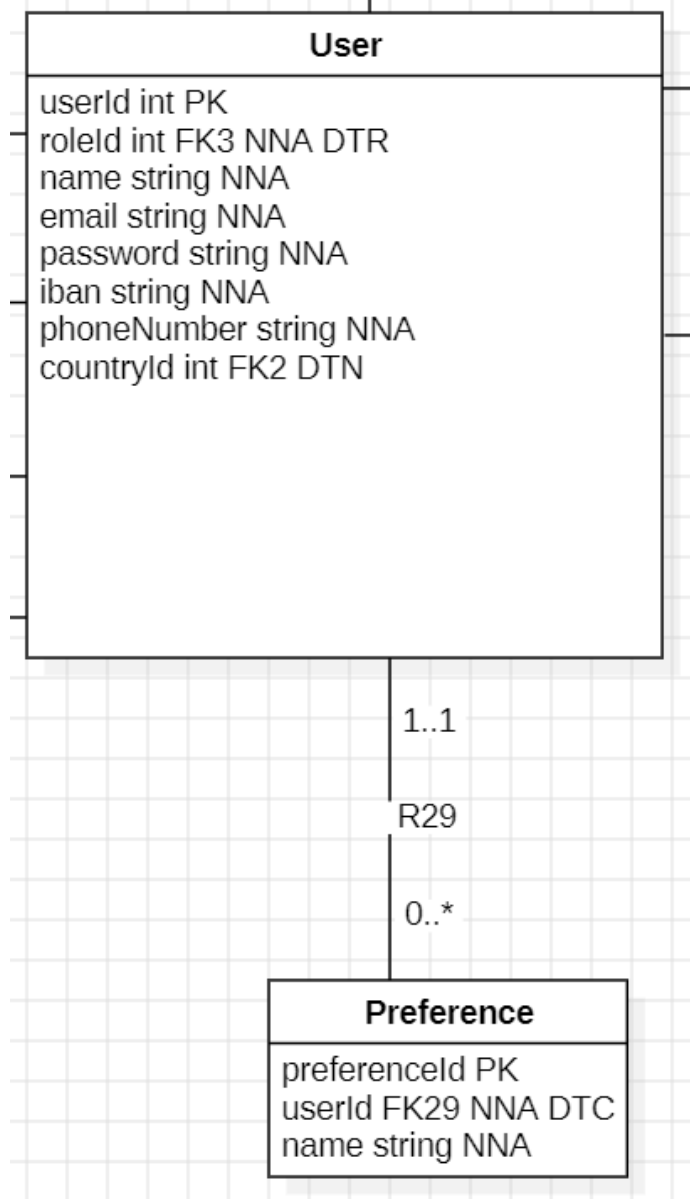
R26 + R27: A trip can have many checkpoints, and a checkpoint can belong to many trips.
tripId FK26 DTC: If a trip is deleted, all checkpoint associations for that trip are deleted as well.
checkpointId FK27 DTC: If a checkpoint is deleted, all trip associations for that checkpoint are deleted as well.



*R28: A checkpoint can belong to a folder, and a folder can have many checkpoints.
folderId FK28 DTC: if a folder is deleted, all checkpoints in this folder are also deleted.*



*R29: A preference belongs to a user, a user can have many preferences
userId FK29 NNA DTC: if a user gets deleted, their preferences also get deleted.*



4. Priority by functionality

Indicate for all functionalities (use cases) which MoSCoW priority should be assigned.

Must Have:

- **Login**
- **Recover Password**
- **Change Password**
- **View Dashboard**
- **View Schedule (Day / Week/ Month)**
- **View Tasks**
- **Manage Personal Tasks**
- **Manage Staff Tasks**
- **Manage Availability**
- **Send Task Reminder**
- **Manage Trips**
- **Manage Checkpoints**
- **Confirm Checkpoints**
- **Manage Planned Meals**
- **Join Planned Meal**
- **Manage Personal Allergies**
- **Manage Personal Receipts**
- **Manage Staff Receipts**
- **Manage Users**

Should have:

- **Update Account Settings**
- **Print Schedule**
- **Print Receipts**

Could have:

- **Request Collaboration**

Won't have:

- **Manage Multiple Households**

5. Backlog

5.1.1. Actor: System User (General)

5.1.1.1. Epic 1: Login

Story: As a System User, I want to log in to the application using my email and password so that I can access my personalized dashboard and features.

5.1.1.2. Epic 2: Update Account Settings

Story: As a System User, I want to update my account details (e.g., name, phone number) so that my contact information remains accurate.

5.1.1.3. Epic 3: Change Password

Story: As a System User, I want to change my current password so that I can keep my account secure.

5.1.1.4. Epic 4: Recover Password

Story: As a System User, I want to request a password recovery link via email so that I can regain access if I forget my credentials.

5.1.1.5. Epic 5: Join Planned Meal

Story: As a System User, I want to view the list of planned meals so that I know what is being served.

Story: As a System User, I want to sign up (join) a planned meal so that the chef knows to prepare a portion for me.

Story: As a System User, I want to remove myself from a planned meal so that food is not wasted if I cannot attend.

5.1.1.6. Epic 6: Manage Personal Allergies

Story: As a System User, I want to add an allergy to my profile so that the chef is aware of my dietary restrictions.

Story: As a System User, I want to view my listed allergies so that I can verify they are correct.

Story: As a System User, I want to remove an allergy from my profile so that outdated restrictions are no longer tracked.

5.1.1.7. Epic 7: View Schedule

Story: As a System User, I want to view the household schedule (day, week, or month) so that I am aware of upcoming events and tasks.

5.1.1.8. Epic 8: Print Schedule

Story: As a System User, I want to print the current schedule so that I can have a physical copy for offline reference.

5.1.1.9. Epic 9: Manage Personal Food Preferences

Story: As a System User, I want to add my food preferences (likes/dislikes) so that meals can be tailored to my tastes.

Story: As a System User, I want to update my food preferences so that they reflect my current tastes.

Story: As a System User, I want to view my recorded food preferences so that I can check what is stored.

5.1.1.10. Epic 10: Manage Notification Preferences

Story: As a System User, I want to select which notifications I receive so that I am only alerted about relevant updates.

Story: As a System User, I want to turn off specific notifications so that I am not disturbed by unnecessary alerts.

5.1.2. Actor: Family Members

5.1.2.1. Epic 11: Manage Trips

Story: As a Family Member, I want to view the list of ongoing and upcoming trips so that I know who is away and when.

Story: As a Family Member, I want to view a detailed list of all upcoming and past trips, including their status and attached files, so that I can maintain an overview of the family's travel schedule.

Story: As a Family Member, I want to update the details of an existing trip (such as changing the dates, notes, or buffer Alert time) so that the itinerary remains accurate if plans change.

Story: As a Family Member, I want to delete a trip that has been cancelled so that the schedule remains clean and avoids confusion.

5.1.2.2. Epic 12: Manage Checkpoints

Story: As a Family Member, I want to add a new Checkpoint to a trip by defining the location, address, and expected arrival date so that the route is clearly defined for the family and security.

Story: As a Family Member, I want to view all checkpoints assigned to a specific trip, ordered by their sequence, so that I can verify the complete route.

Story: As a Family Member, I want to edit the details of a checkpoint (e.g., changing the location address or expected arrival time) so that the travel logistics are up to date.

Story: As a Family Member, I want to remove a checkpoint from a trip so that stops that are no longer part of the itinerary are deleted from the system.

5.1.2.3. Epic 13: Confirm Checkpoints

Story: As a Family Member, I want to confirm arrival at a specific checkpoint so that the staff and family know I have arrived safely.

5.1.3. Actor: The Andersons (Heads of Household)

5.1.3.1. Epic 14: View Invoices

Story: As "The Andersons", I want to view a list of all submitted invoices so that I can oversee household expenses.

Story: As "The Andersons", I want to filter invoices by status (paid/unpaid) so that I can focus on pending payments.

5.1.3.2. Epic 15: Approve Invoices

Story: As "The Andersons", I want to approve a pending invoice so that the payment process can proceed.

Story: As "The Andersons", I want to mark an invoice as "Paid" so that our financial records are accurate.

5.1.4. Actor: Staff Member

5.1.4.1. Epic 16: Manage Personal Tasks

Story: As a Staff Member, I want to create a personal task so that I can remember my own to-do items.

Story: As a Staff Member, I want to view my list of personal tasks so that I know what I need to complete today.

Story: As a Staff Member, I want to update the status of a task (e.g., to "Completed") so that I can track my progress.

Story: As a Staff Member, I want to delete a personal task so that I can remove items created in error.

5.1.4.2. Epic 17: Request Collaboration

Story: As a Staff Member, I want to request collaboration from another staff member on a task so that we can work on it together.

5.1.4.3. Epic 18: Manage Personal Invoices

Story: As a Staff Member, I want to upload a new invoice/receipt so that I can be reimbursed for household expenses.

Story: As a Staff Member, I want to view the status of my submitted invoices so that I know if they have been approved or paid.

Story: As a Staff Member, I want to edit invoice details (if unapproved) so that I can correct mistakes before approval.

5.1.4.4. Epic 19: Manage Personal Availability

Story: As a Staff Member, I want to mark periods of unavailability (e.g., time off) so that the Admin knows when I am not free to work.

- **Story:** As a Staff Member, I want to view my registered unavailability periods so that I can confirm my schedule.
- **Story:** As a Staff Member, I want to delete an unavailability period so that I can open up my schedule if my plans change.

5.1.5. Actor: Chef

5.1.5.1. Epic 20: View Planned Meals

Story: As a Chef, I want to view all planned meals for the week so that I can prepare the necessary ingredients.

Story: As a Chef, I want to see which users have subscribed to a specific meal so that I cook the correct number of portions.

5.1.5.2. Epic 21: Confirm Planned Meal

Story: As a Chef, I want to confirm that a planned meal has been prepared so that the household knows dinner is ready.

5.1.5.3. Epic 22: View All Allergies

Story: As a Chef, I want to view a consolidated list of all user allergies so that I can ensure no dangerous ingredients are used in meals.

5.1.6. Actor: Admin

5.1.6.1. Epic 23: Manage Meal Plan

Story: As an Admin, I want to create a new planned meal for a specific date and time so that the household knows the schedule.

Story: As an Admin, I want to view the calendar of planned meals so that I can ensure there are no gaps in the meal schedule.

Story: As an Admin, I want to update details of a planned meal (e.g., menu description) so that information remains accurate.

Story: As an Admin, I want to delete a planned meal so that cancelled dinners are removed from the schedule.

5.1.6.2. Epic 24: Manage Staff Invoices (Receipts)

Story: As an Admin, I want to view all invoices submitted by staff so that I can audit expenses.

Story: As an Admin, I want to update the category of a receipt so that expenses are allocated to the correct budget.

Story: As an Admin, I want to delete duplicate or erroneous receipts so that the financial records are clean.

5.1.6.3. Epic 25: Print Receipt

Story: As an Admin, I want to print a specific receipt/invoice so that I have a paper trail for accounting.

5.1.6.4. Epic 26: Manage Staff Tasks

Story: As an Admin, I want to create a task and assign it to a staff member so that work is distributed efficiently.

Story: As an Admin, I want to view all staff tasks so that I can monitor the household's workload.

Story: As an Admin, I want to update task details (e.g., deadline, priority) so that staff have the latest instructions.

Story: As an Admin, I want to delete a task so that unnecessary work is removed from the list.

5.1.6.5. Epic 27: Manage Users

Story: As an Admin, I want to create a new user account (Staff or Family) so that new members can access the system.

Story: As an Admin, I want to view a list of all registered users so that I can see who has access.

Story: As an Admin, I want to update a user's role or details so that permissions remain correct.

- **Story:** As an Admin, I want to deactivate or delete a user account so that former staff no longer have access.

5.1.7. Actor: Time (System)

5.1.7.1. Epic 28: Send Task Reminder

Story: As the System (Time), I want to send an automatic reminder to users about upcoming tasks so that deadlines are not missed.