2025

Vacation Tracking System - VTS

DOCUMENTATION

ABDALLA KHALIFA

Vision

The main goal of this system is to improve the internal process of the organization respecting the time it takes to manage vacation time, providing individual employees, the ability to manage their own vacation time, sick leave and personal time off without knowledge in company policy.

Actors & Activities

We have four actors in the scene:

1- **Employee**: Manage Time [Create new requests].

2- **Manager**:

   i. Approve requests.

   ii. Award time.

3- **HR**:

   i. Edit employee record.

   ii. Manage locations.

   iii. Override leave records.

   iv. Manage leave categories [sick leave etc..].

4- **System Admin**: Backup system logs.

Functional Requirements

- Rule based system.
- Enables manager to approve vacation requests [optional].
- E-mail notification [Manager – Employee].
- System should use the portal's single – sign – on mechanisms for all authentications.
- Activity logs for all transactions [Auditing].
- HR and system admin have the ability to override rules & Auditing.
- Enables manager to directly award personal leave time according to system roles.
- Integration services.

Non – Functional Requirements

- **Usability**: The system must be easy to use giving the user a smooth and simple experience.
- **Scalability**: The system should have the ability to scale and add new features and integrate with systems related to the facility.

Constraints

- The system uses the company existing hardware and middleware [legacy hardware].
- The system is extension to the existing intranet portal system.
- The system uses the portal's single-sign-on mechanism for all authentications.
- All system transactions are saved in the system logs.
- User experience.

Business Roles

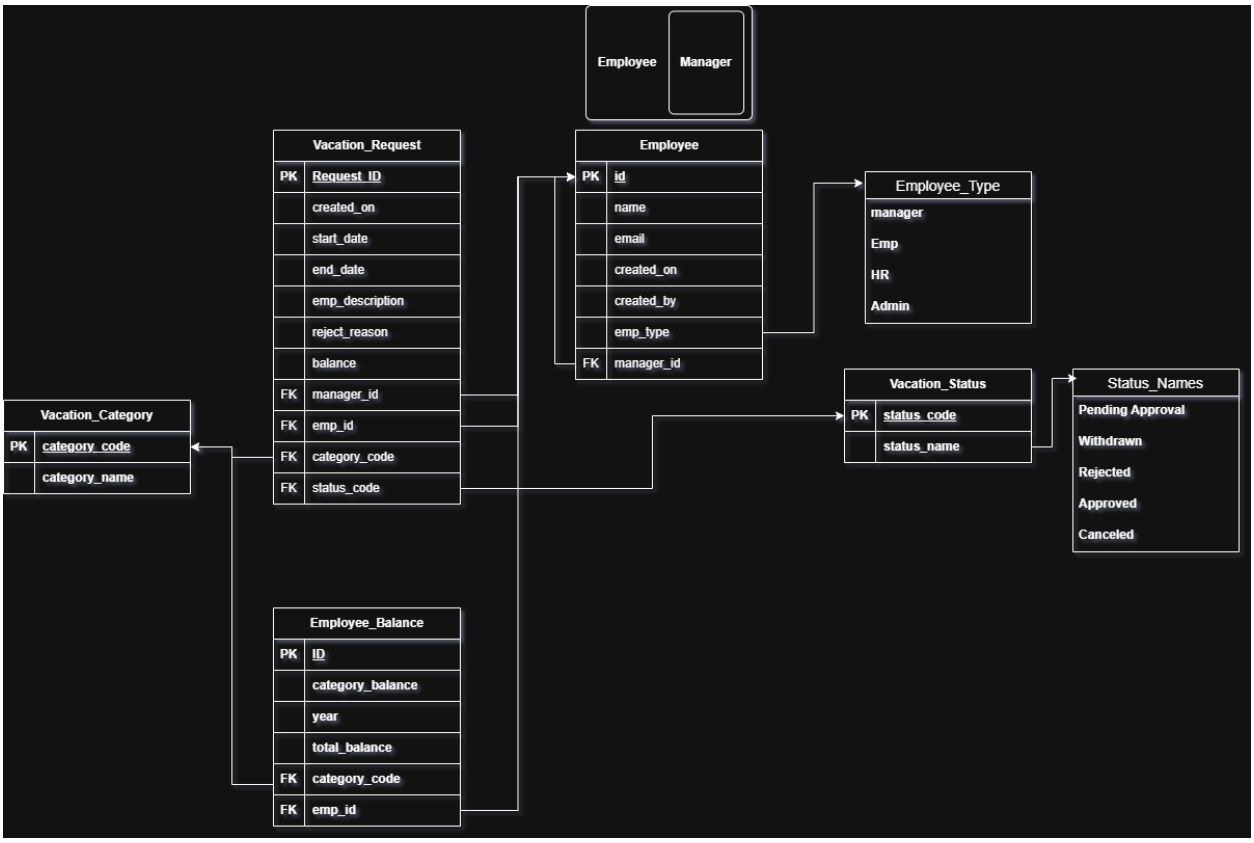
- All employees work eight hours a day.
- Each employee's vacation time requests are subject to the restrictions of each employee's primary work location in addition to overall company policies and restrictions.
- Vacation requests validation rules are defined and owned by the HR department.
- An employee can't take more than x consecutive days of leave for Y type of grant.
- Vacation time of type X cannot be taken when directly adjacent to a company or location – specific holiday.
- Vacation time may not be granted when there is only x number of employees scheduled to work from a predefined list of Y employees (there is short of employees count).
- Vacation time may not be granted on specific dates.
- Vacation time of a certain type is limited to certain days of the week.

Note: All specific constraints are in the company policy maintained by the HR department.

Assumptions

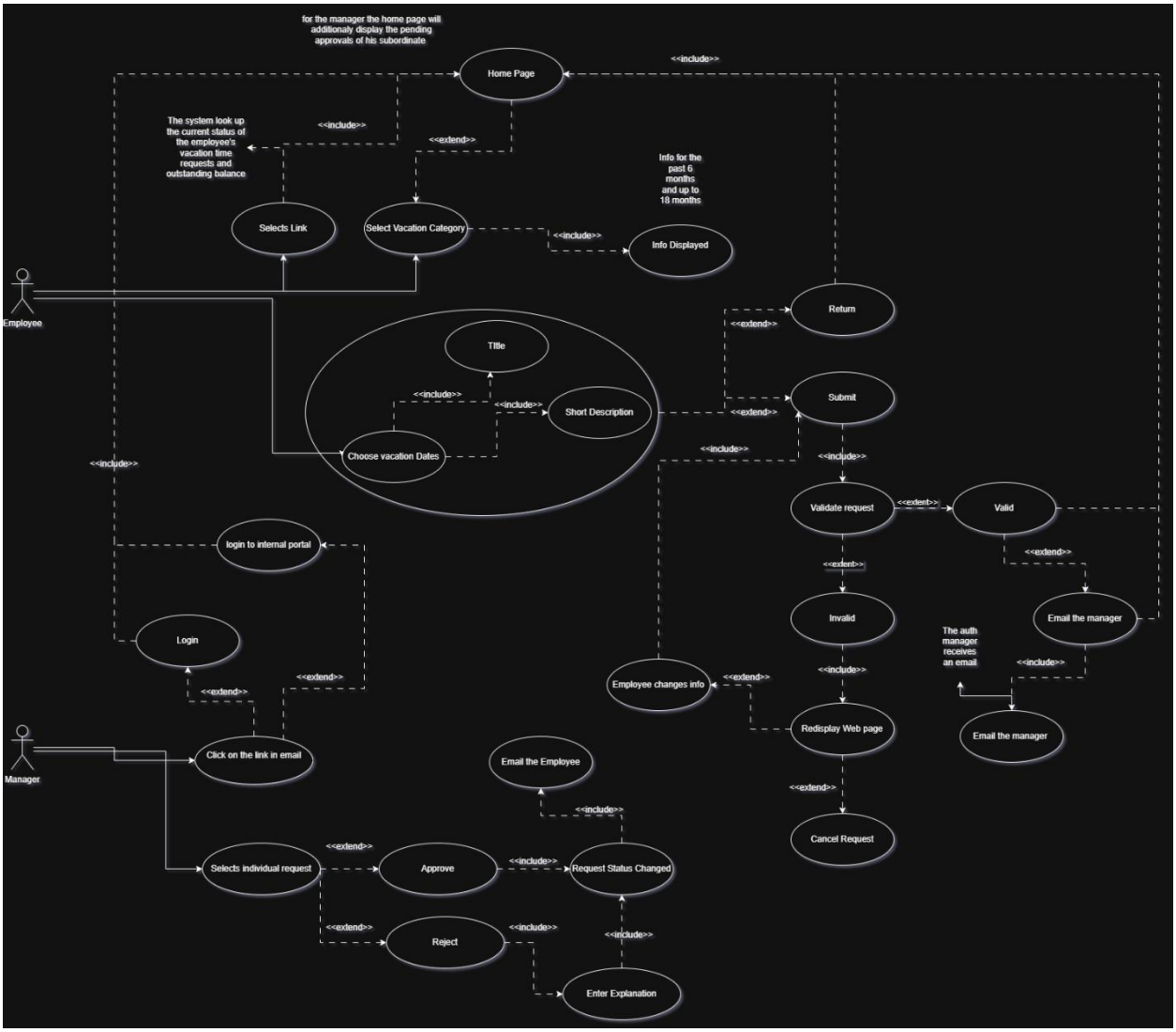
- The company's existing software is scalable and have the ability to integrate with new made extensions.
- The company's infrastructure is sufficient and supports our needs.

Data Model

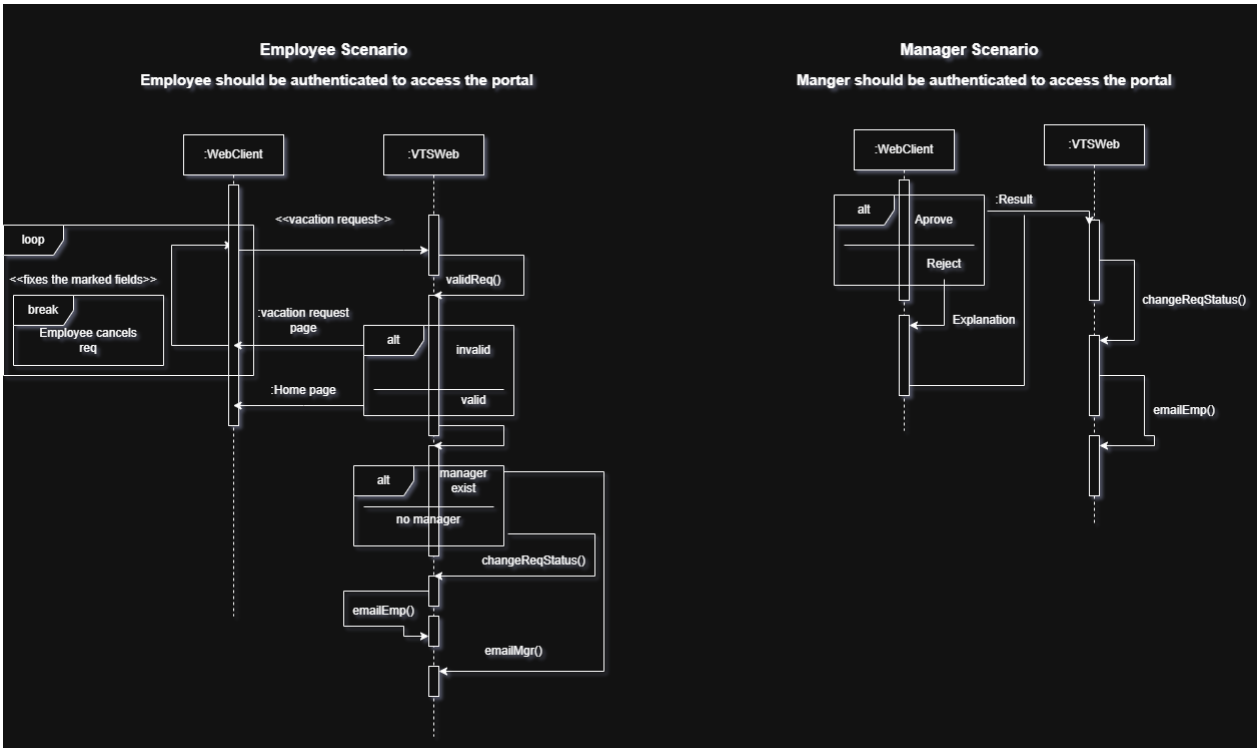

Employee
Manager
Vacation_Request
PK Request_ID
created_on
start_date
end_date
emp_description
reject_reason
balance
FK manager_id
FK emp_id
FK category_code
FK status_code
Employee
PK id
name
email
created_on
created_by
emp_type
FK manager_id
Employee_Type
manager
Emp
HR
Admin
Vacation_Category
PK category_code
category_name
Vacation_Status
PK status_code
status_name
Status_Names
Pending Approval
Withdrawn
Rejected
Approved
Canceled
Employee_Balance
PK ID
category_balance
year
total_balance
FK category_code
FK emp_id

Manage Time Use Case

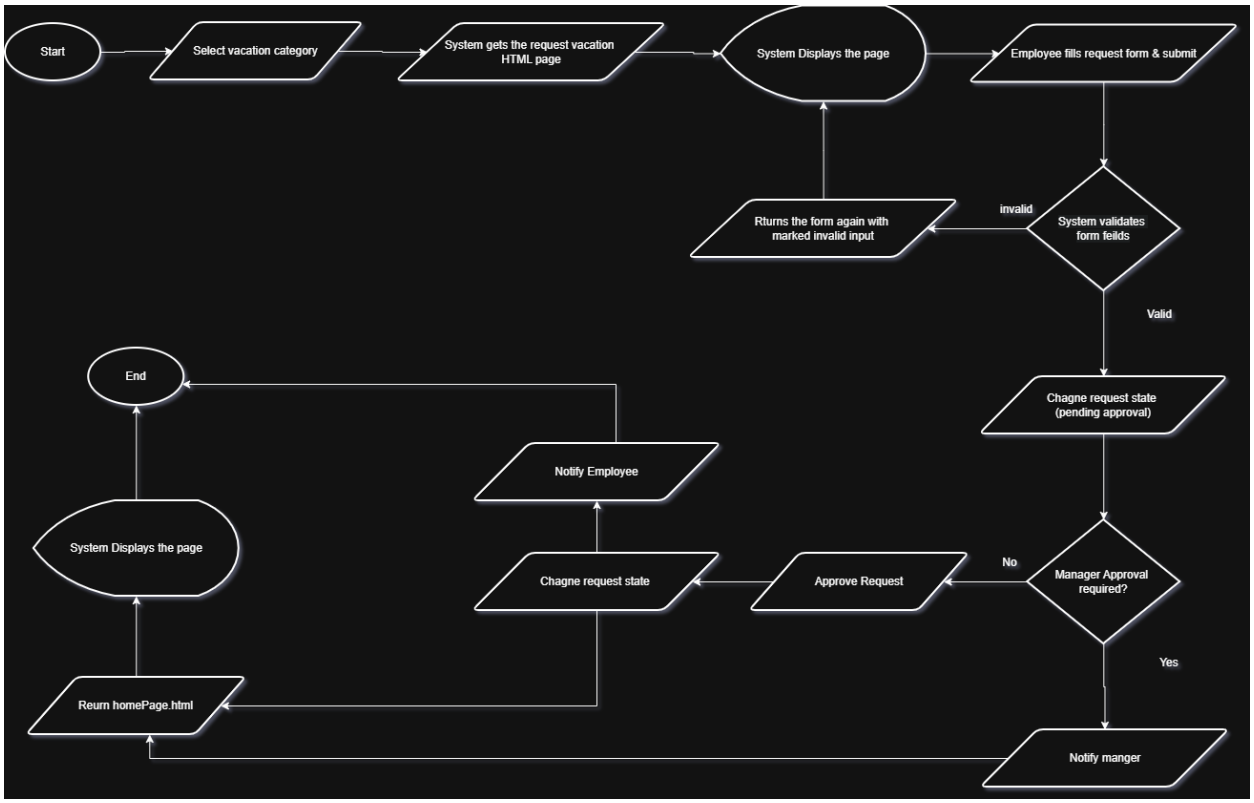
Use Case Diagram:



Sequence Diagram:



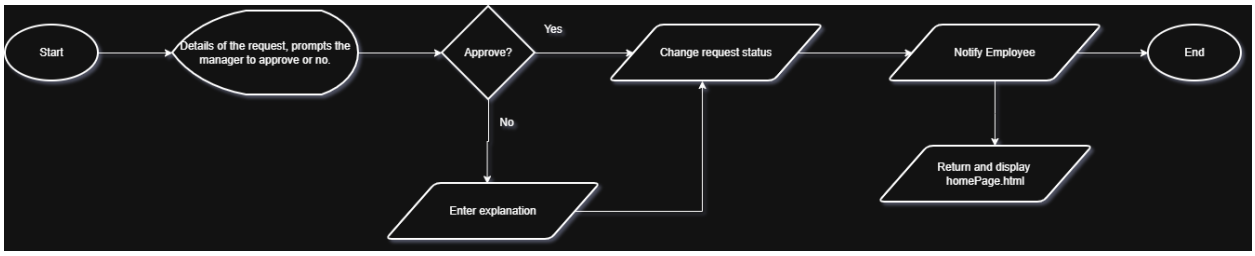
Create New Request (Flowchart):



Preconditions:

1. The employee should be authenticated by the portal and have the privileges to manage his / her own vacation time.
2. System already returned and displayed the home page.

Manager Handles Employee Request:



Preconditions:

- The manger received an email about a new request and clicked the embedded link in the email.
- The manager is authenticated by the portal to manage his personals requests.
- The VTS displayed the separate section listing requests pending approval.
- The manager selected individual request.

Pseudocode For New Request Scenario

```
Function void createReq(Request: request){

        If (!isValidRequest(request))
                Throw Error;

        request.status=”pending approval”;
        async notifyManager(request);
        request.save(); //DB – we also have a transaction to change the emp balance.

}
Function Boolean isValidRequest(Request: req){

        if(!isValidReqDateFromAndTo(req.from, req.to));
                return false;

        if(!isValidCategoryBalance(req.categoryCode, req.empID, req.balance))
                return false;

        return true;
}
Function Boolean isValidReqDateFromAndTo (from, to){
        //Handle with DateTime APIs
}
Function Boolean isValidCategoryBalance (categoryCode, empID, reqBalance){
        Int empBlanace=getEmployeeBalance(empID, categoryCode); //DB call
        Return empBalance > reqBalance;
}
```

Pseudocode For New Request Approval Scenario

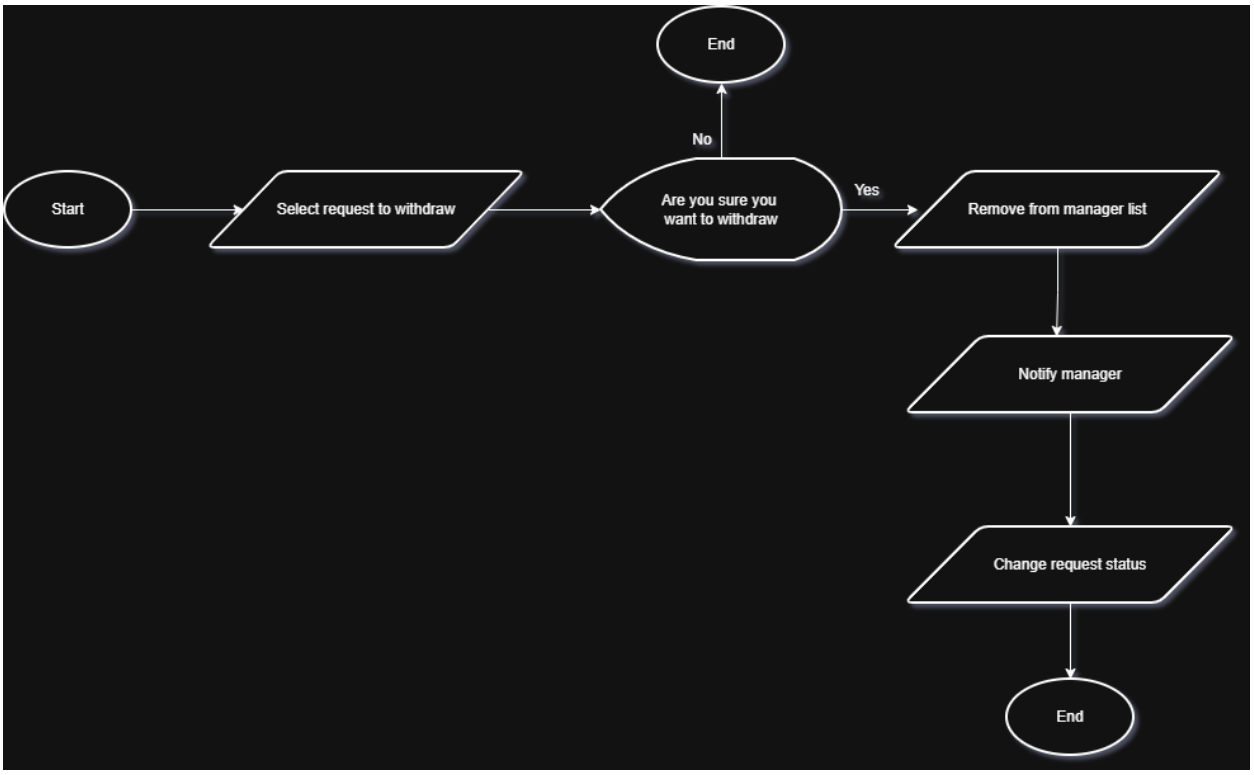
Preconditions: Manager is authenticated by the portal and reviewed the request.

```
Function changeRequestStatus(managerSelection, request){
        Request.status=managerSelection;
        manageRequest(request);
}
Function manageRequest(Request: request){
        Async notifyEmployee(request);
        Request.updateStatus(); //DB

}
```

Withdraw Request Use Case

Flowchart:

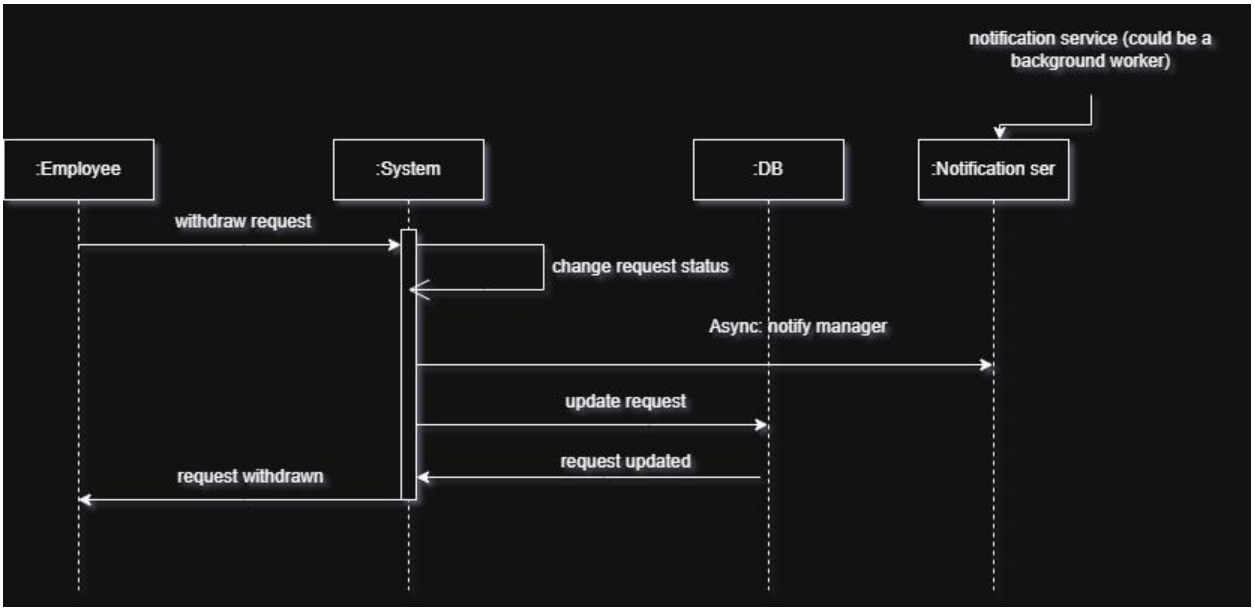


Preconditions: Employee already made a request and the manager didn't take any actions yet. The employee is authenticated by the portal and navigated to the VTS homepage. The VTS displays summary of requests for previous 6 months and 18 months in the future and gives the employee the ability to withdraw his pending requests.

Pseudocode:

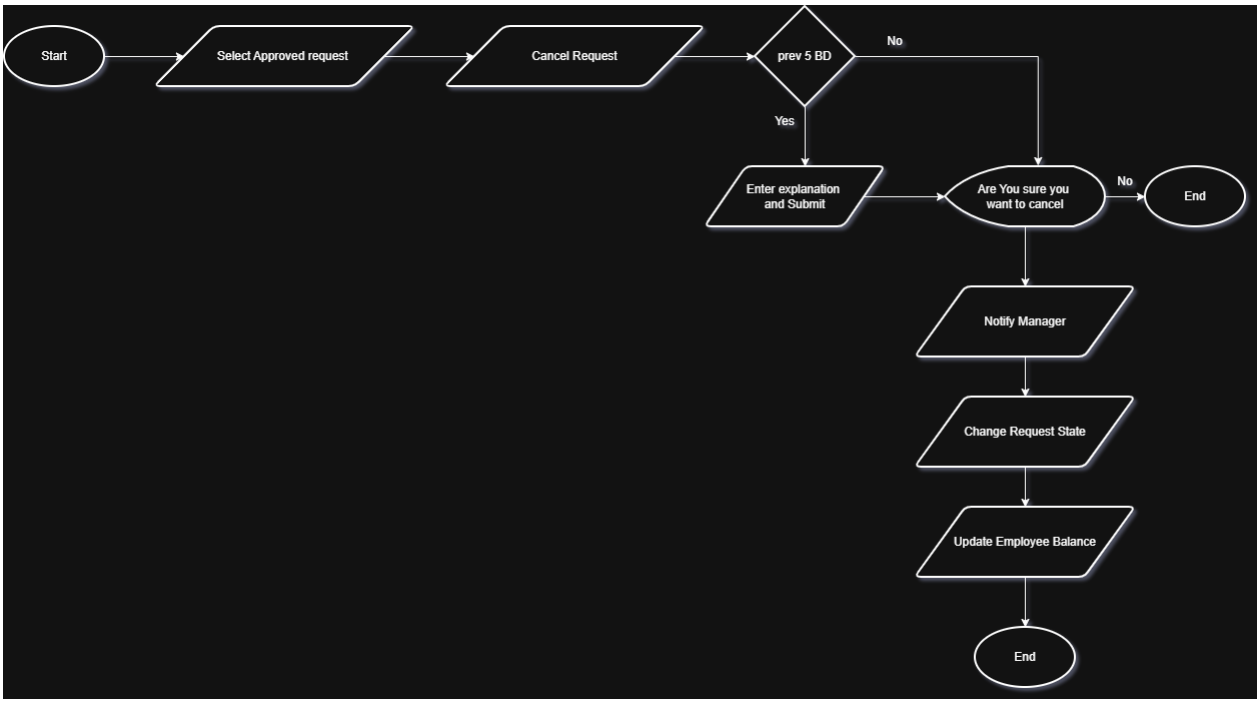
```
 Function changeRequestStatus(request){
        Request.statu s= ”Withdrawn”;
        manageRequest(request);
}
Function manageRequest(Request: request){
        Async notifyManager(request);
        Request.updateStaus(); //DB: setting the status to withdrawn will-
                               // automatically remove it from the manager list
}
```

Sequence Diagram:



Cancel Approved Request

Flowchart:

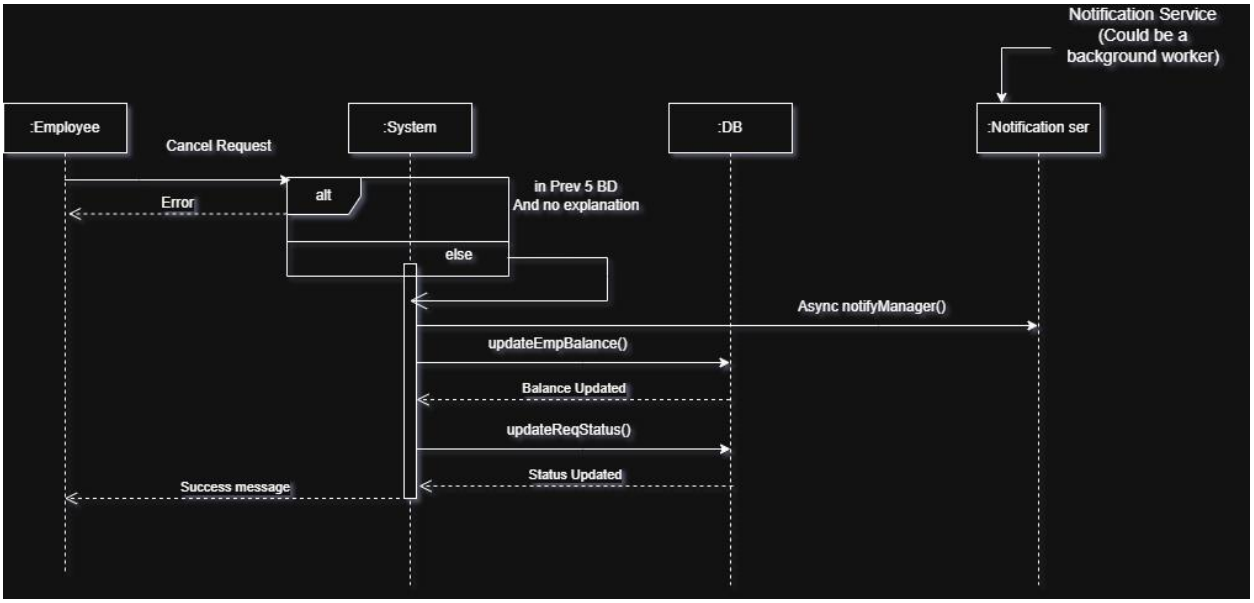


Preconditions: The request already approved by the manager. The request should be in the future or within the previous 5 working days (he might be forgotten to cancel it previously).

Pseudocode:

```
Function cancelRequest(Request: request){
        If(isInPrev5BD(request)){
                If(explanationEmpty())
                        Throw Error;
        saveCancelledRequest(request.id, explanation) //DB
        }
        Async notifyManager(request);
        Reuquest.status=”canceled”;
        returnEmpBalance(request.balance, request.empId); //DB
        Request.updateStatus(); //DB
}
```

Sequence Diagram:



Edit Pending Request

Flowchart: