



Creating an app for the Out of Office solution

Technical specification

Contents

Creating an app for the “Out of Office” solution.....	2
General functionality overview.....	2
“Out of Office” solution directories.....	5
Employee	5
Leave Request.....	6
Approval Request.....	7
Project	7
Roles in the system (***).....	8
Task setting	8
Additional tasks	8

Creating an app for the “Out of Office” solution

General functionality overview

1. The HR manager opens the system and has to see the ability to navigate to the list of employees, projects, leave requests, approval requests.

With a list of employees, the HR manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by name for table rows; – (**)
- add/update/deactivate employees in the list.

With a list of approval requests, the HR manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a request – a new form opens with details.
- approve the request – related leave request is updated with the corresponding status, and employee absence balance is recalculated;
- reject the request – related leave request is updated with the corresponding status. Write a comment explaining that rejection should be possible.

With a list of leave requests, the HR manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a request – a new form opens with details.

With a list of projects, the HR manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a project – a new form opens with details.

2. The Project manager opens the system and has to see the ability to navigate to the list of employees, projects, leave requests, approval requests.

With a list of employees, the Project manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by name for table rows; – (**)
- open an employee – a new form opens with details.
- assign an employee to projects.

With a list of approval requests, the Project manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a request – a new form opens with details.
- approve the request – related leave request is updated with the corresponding status, and employee absence balance is recalculated;
- reject the request – related leave request is updated with the corresponding status. Write a comment explaining that rejection should be possible.

With a list of leave requests, the Project manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a request – a new form opens with details.

With a list of projects, the Project manager can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a project – a new form opens with details
- add/update/deactivate projects in the list.

3. The company employees opens the system and has to see the ability to navigate to the list of their projects and leave requests.

With a list of leave requests, the employee can:

- sort table rows using sorting in the column headers,
- use filters for table rows; – (*)
- use search by request number for table rows; – (**)
- open a new and existing request – a form opens, and employee can create and update the request.
- submit the request – the status is updated to “Submitted,” and a new approval request is created for the responsible HR Manager and Project Managers responsible for employee’s projects;

- cancel the request – the status is updated to “Canceled,” and approval requests are canceled if they already exist.

Basic level:

The roles of HR manager, Project Manager and Employee may not be separated, but different views can be made to open the corresponding lists for the tasks listed above.

Additional level – (*):**

The roles of HR manager, Project manager, and employee should be separated. The roles must be allocated and have different access levels. The HR manager has full access to the list of their employees and approval requests and read-only access to leave requests and projects related to their employees. The Project manager has full access to the list of their projects and approval requests and read-only access to leave requests and employees related to their projects. The employee has full access to the list of their own leave requests, read-only access to approval requests related to their own leave request, and read-only access to their projects.

Complexity levels:

- without marks – basic required level.
- (*) additional task with complexity level – 1.
- (**) additional task with complexity level – 2, can be performed without level 1.
- (***) additional task with complexity level – 3, can be performed without levels 1 and 2.

“Out of Office” solution directories

The “Out of Office” solution contains a set of directories that must be created.

Lists:

1. Employees (/Lists/ Employees/)
2. Leave Requests (Lists/Leave Requests/)
3. Approval Requests (opened from the Leave Request form by an employee(***) and from Lists/Approval Requests/ by HR manager and Project Manager)
4. Projects (Lists/ Projects/)

Employee

No.	Column	Type	Required	Comment
1.	ID (non-editable)	INT PK	yes	Unique record identifier
2.	Full Name	Selection	yes	One-line field
3.	Subdivision	Selection	yes	Single choice from the “Subdivision” list
4.	Position	Selection	yes	Single choice from the “Position” list
5.	Status	Selection	yes	Active/Inactive
6.	People Partner	INT FK	yes	Single choice from the “Employee” table with “HR Manager” position
7.	Out-of-Office Balance	Digit	yes	Number of available day-offs
8.	Photo – (*)	Files	no	Item photo

Leave Request

No.	Indicator	Type	Entry required	Comment
1.	ID (non-editable)	INT PK	yes	Unique record identifier
2.	Employee	INT FK	yes	Single choice from the “Employee” table
3.	Absence Reason	Selection	yes	Single choice from the “Absence Reason” list
4.	Start Date	Date	yes	Date field
5.	End Date	Date	yes	Date field
6.	Comment	Text	no	Multiline field
7.	Status – (non-editable)	Selection	yes	Single choice from the “Status” list. It is updated based on clicking the Submit/Cancel button and by the result of the Approval Request. The default value is <i>New</i> .

Approval Request

No.	Indicator	Type	Entry required	Comment
1.	ID (non-editable)	INT PK	yes	Unique record identifier.
2.	Approver	INT FK	yes	Single choice from the “Employee” table
3.	Leave Request	INT FK	yes	Single choice from the “Leave Request” table
4.	Status – (non-editable)	Selection	yes	Single choice from the “Status” list. Is updated upon clicking Approve/Reject Request. The default value is <i>New</i> .
5.	Comment	Text	no	Multiline field

Project

No.	Indicator	Type	Entry required	Comment
1.	ID (non-editable)	INT PK	yes	Unique record identifier
2.	Project Type	Selection	yes	Single choice from the “Project Type” list
3.	Start Date	Date	yes	Date field
4.	End Date	Date	no	Date field
5.	Project Manager	INT FK	yes	Single choice from the “Employee” table with “Project Manager” position
6.	Comment	Text	no	Multiline field
7.	Status – (non-editable)	Selection	yes	Active/Inactive

Roles in the system (***)

User name	Description of main tasks
Employee	<ul style="list-style-type: none">• Creates a leave request
HR Manager	<ul style="list-style-type: none">• Manages the list of employees• Approves/rejects requests
Project Manager	<ul style="list-style-type: none">• Manages the list of projects• Approves/rejects requests
Administrator	<ul style="list-style-type: none">• Grants access rights• Manages all data

Task setting

1. Create an SQL database with the above tables and fields.
2. Create an application that will interact with the database.
3. Add functionality for working with the created database in the form of various roles in the system. Reproduce the conditions necessary for users to work with the system.
4. Provide the result of the completed task, which includes the following:
 - SQL queries that were used to create the database. Schematic display of table relations in the database with a description;
 - code of the developed application – should be provided in GitHub;
 - a file with screenshots of the finished application with comments.
5. Programming language – arbitrary.

Completing all points is not necessary to successfully pass the test task.

Additional tasks

Additional tasks are indicated by (*), (**) and (***). In accordance with the number of *, the tasks become more difficult. Each task with * can be completed separately, without having to complete the task of the previous level. Please note that when completing the (***) task you will need to process entities in the database, so in this case, when submitting the test task, you will need to provide both diagrams of the database structures and the corresponding creation scripts.

If necessary, the structure of entities can be created at the discretion of the task performer, but this must be reflected in the diagrams.