**Session 2 – Inventory Management**

**Duration: 150 minutes**

# 1 Instructions

This document contains the special tasks for the 2nd session. We advise you to read this document carefully and to get an overview over the provided media like attached images, appendices and text.

The maximum time to complete as many tasks as possible is **150 minutes**. **Within this time,** you must read and understand all tasks and to store the results **into the folder** “***Session-2-yourname***” onto the desktop of your computer.

This is the first session – that means you will write the basis for the others. You are allowed to use all components, source code and windows of this session for the following sessions of the competition. Sometimes it makes sense to use parts of it also for other tasks in the future.

Please for every session create a file containing a short user guide and a technical description of how to configure and start up the application.

Make sure that you follow the provided style guide throughout all parts of the system.

Make sure that you provide appropriate validation and error messages throughout all parts of the system.

Make sure that all relevant buttons/links are working at the end of the session.

Make sure that you use appropriate naming conventions for all parts of the system as needed.

Make sure that your work is on the state of the art of Skill 09.

# 2 Contents

Session 2 of this Test Project consists of the following documentation/files:

1. AEC2024-Skill09-Session 2.pdf
2. Application Structure Session 2.pdf
3. Session2.sql

ө

# 3 Project

## 3.1 Introduction

The **Event** **Agency** **NeuBrandenburg** (**EANB**) is a modern enterprise which organizes, manages and offers cultural events in Neubrandenburg and its environment.

This agency needs a mobile app for event organizers to manage the inventory for events and to rent and lease items according to the search/offer principle.

## 3.2 The Task

The app uses a central server for data management. The database is partially given. Event organizers must be able to request and offer items required for carrying out the events.

# 4 Application Structure

The application consists of two main screens: the ***Login-Screen*** andthe ***Main-Screen***.

All these screens are connected using special buttons or functions respectively.

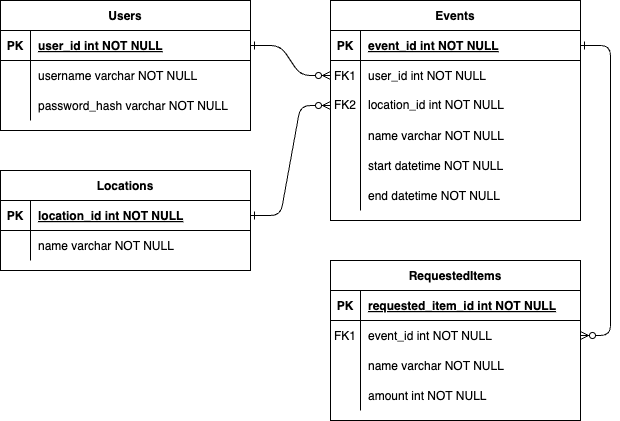
# 5 Tasks

## 5.1 Database and API Server

**Quality tested** Database, Programming, API, DB Access

**Work requested** General

* The database should be implemented on the server
* Execute Session2.sql to import the given parts of the database
* Read the following chapters carefully because they contain important information about the required structure
* Extend the database to support all the required features and attain the ***highest possible and feasible normal form level***
* Use appropriate data types
* Passwords and other security related information should be encrypted
* The backend should support all operations required by the mobile app
* Refer to the ERD below for the database structure



## 5.2 Login Screen

**Quality tested** Programming, UI Design, DB Access

**Mockup** to be useful design

**Work requested** General

* Create the Login Screen
* The user must enter a *username* and *password*
* When the login button is pressed, the credentials are checked against the user table by the server
* The password column of the user table contains salted SHA256 hashes of the password These are generated in the following way:
  + 1. Generate a salt consisting of 8 random characters
    2. Append the salt to the password and decode both to bytes using UTF8
    3. Hash the bytes using SHA256 and encode the bytes using base64
    4. The value in the database is formed by appending the encoded hash to the salt
  + If the user entered valid credentials proceed to the main screen
  + If the user entered invalid credentials display an appropriate error message
  + If the server is not available signal the user that the device is offline
  + You can use these accounts for testing:

| Username | Password |
| --- | --- |
| j.doe | ABC123a! |
| k.omar | House4258+ |

## 5.3 Main Screen

**Quality tested** Programming, UI Design

**Mockup** to be useful design

**Work requested** General

* Create the screen
* The screen consists of three tabs: *Events*, *Requests* and *Offers*
* The user should be able to switch between tabs by swiping left or right or by selecting a tab in a bottom sheet

## 5.4 Events tab

**Quality tested** Programming, UI Design, DB Access

**Mockup**  to be useful design

**Work requested** General

* Create the events tab
* All events organized by the user that have not ended yet are shown in a list and sorted by the start time in ascending order
* The *name*, *place*, *date*, *time* and *requested inventory* are shown for each event
* The *requested inventory* consists of multiple items that are displayed as a checklist
* The *name* and *requested amount* of each item needs to be displayed, and the checkbox is checked if there is a reservation for the requested items
* When the user clicks on an unfulfilled request a dialog is shown that allows the user to accept an offer
* The dialog shows all offers (including offers by the user himself) that can satisfy the request
* An offer satisfies the request if it provides the number of items requested or more and is not already reserved during the timespan from 12h before the start of the event to 12h after the end of the event
* The user can filter the offers by name using a search field
* The user can select an offer to create a reservation for the timespan from 12h before the start of the event to 12h after the end of the event for all items of the offer, even when the amount of the offer exceeds the request
* When the user clicks on a fulfilled request a dialog is shown that shows the reserved offer, the name of the user responsible for the offer and the timespan of the reservation
* If the server is not available signal the user that the device is offline and allow him to reload the page

## 5.5 Requests tab

**Quality tested** Programming, UI Design, DB Access

**Mockup**  to be useful design

**Work requested**  General

* Create the requests tab
* All items requested by other users are listed in ascending alphabetical order
* The *amount, item name* and *reservation timespan* from 12h before the start of the event to 12h after the end of the event are shown for each request
* When the user clicks on a request a dialog is opened
* The dialog shows all offers the user created that can satisfy the request
* The user can select an offer to fulfil the request
* The user can also create a one-time offer to fulfil the request without an existing matching offer
* If the server is not available signal the user that the device is offline and allow him to reload the page

## 5.6 Offers tab

**Quality tested** Programming, UI Design, DB Access

**Mockup**  to be useful design

**Work requested**  General

* Create the offers tab
* All items offered by the user are listed in ascending alphabetical order
* The *amount, item* name and *reservations* are shown for each offer
* The reservations consist of the *timespan* of the reservation and the *username* of the user that reserved the items
* A dialog for editing is shown when clicking on an offer
* The dialog allows the user to update the *name* and *amount*
* There is a button for removing the offer
* Existing reservations are not affected by updating/deleting the offer
* There is a button that opens a dialog for creating offers
* The user must enter a *name* and *amount*
* If the server is not available signal the user that the device is offline and allow him to reload the page

## 5.7 Documentation / Notes

**Quality tested** Documentation

**Work requested**

* Please create a file containing a short user guide and a technical description of how to configure the server, how to get access to your database and how to start up the application
* In this file you may also leave *notes* that could be useful for the evaluation. *Notes* in this document are not part of the evaluation process