# Purpose

This document aims at giving some common ground requirements to build the demo application.

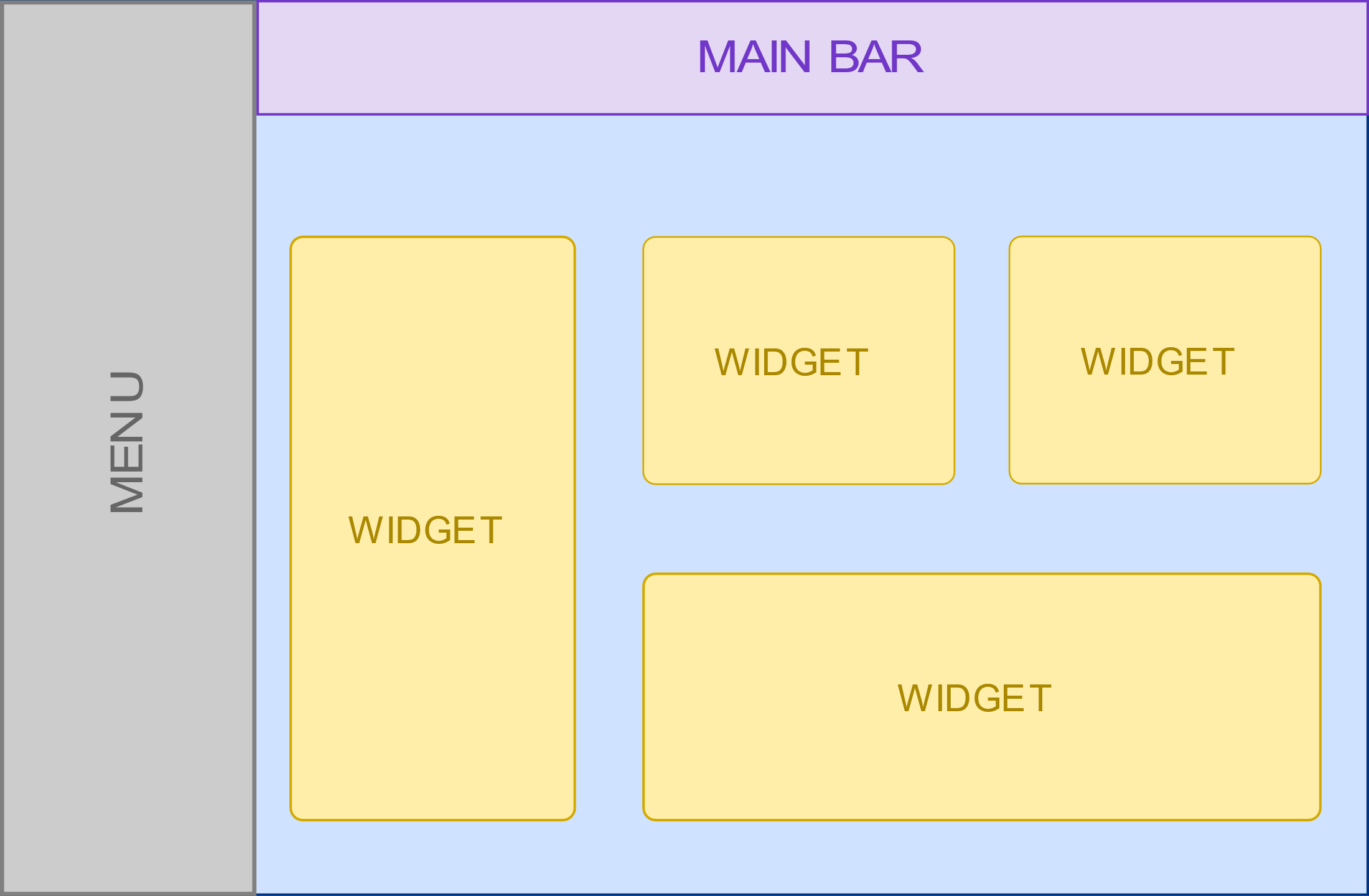
Anything mentionned here is mandatory but one can add more functionalities.

# Screen 1 – Home screen

## Purpose

Welcome the user into the application and display some informations and indicators.

## Zoning



### Components

#### Menu

Presented using a hierarchy, it should contain the following items :

* Home
  + My Activity
    - To Do
      * My Actions
      * Actions For my Groups
      * Incidents On Hold
      * KB Articles to Review
    - Knowledge
      * KB Articles
      * Dell Support Wiki
      * SAP Knowledge Base
    - Asset Management
      * Discovery
      * Consumable Stock
      * Active Equipment
    - More
      * 1
      * 2
  + History
    - 20 lasts
    - Content Records
  + Analytics
    - Reporting
    - Dashboards
    - Business Intelligence
  + References
    - Employees
    - Location
    - Department

The highlighted items are the only active links and lead to the 2 other screens :

* Incidents On Hold leads to Incidents screen.
* Employees leads to Employees screen.

#### Main bar

Displays a global search bar (functionality not to fully implement), the connected user and a disconnect button.

#### Widget

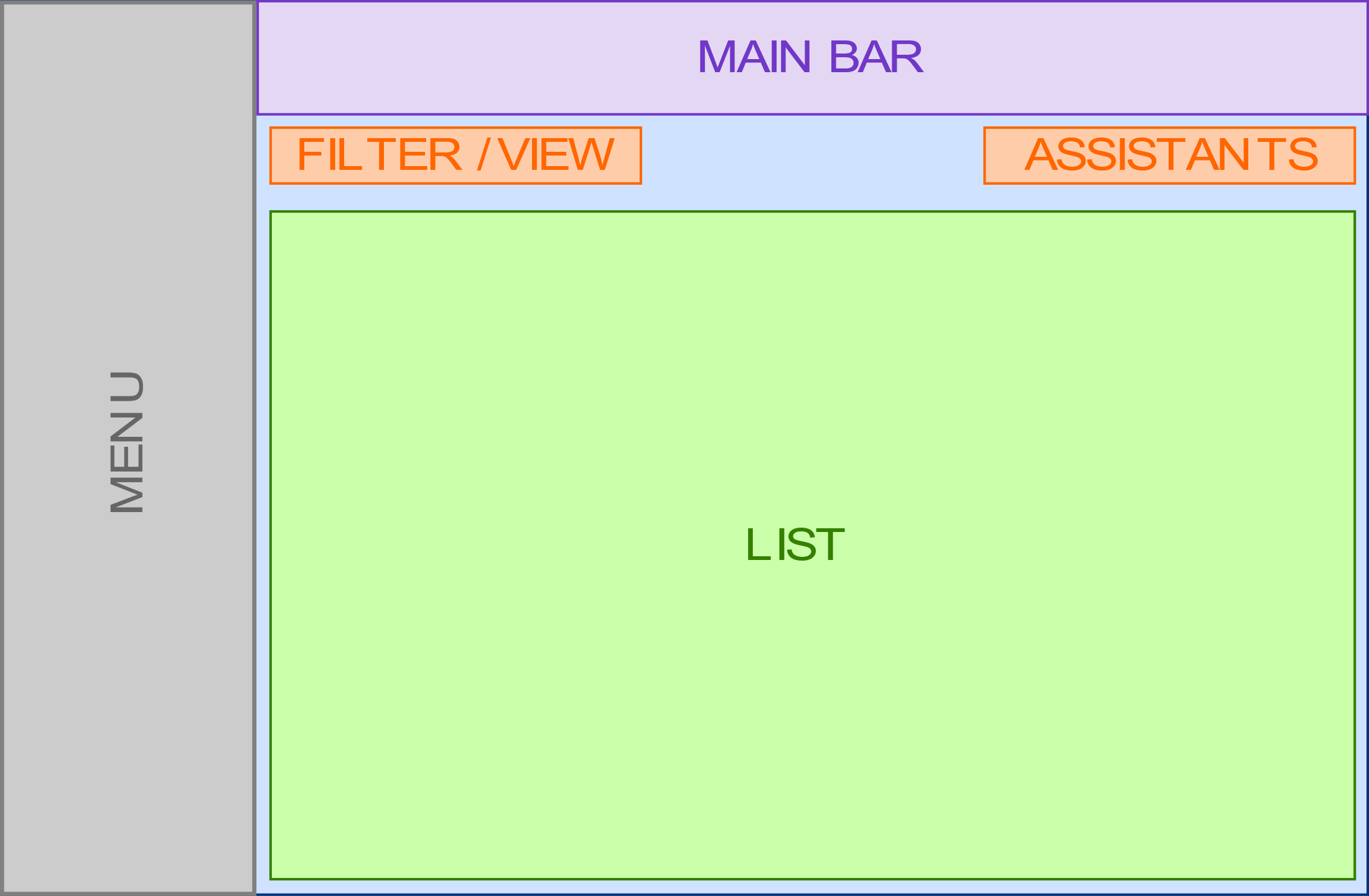
Some widget demonstrating various integrations, such as graphics (if possible). The corresponding code should illustrate the reusability of the widget code and that it is possible to exploit some inheritance mecanism between widgets.

# Screen 2 – Incidents List Screen

## Purpose

This screen requests the EVSM API to retrieve a list of incidents and display it.

## Zoning



### Components

#### Menu

Same as Home Screen

#### Main bar

Same as Home Screen

#### Filter / View

Displays the current filter and view (functionality not to fully implement).

#### Assistants

Display some buttons as shortcuts to frequently used assistants.

A button on the right shows the list of the remaining assistants.

#### List

Displays the records retrieved from EVSM.

The list should be sortable.

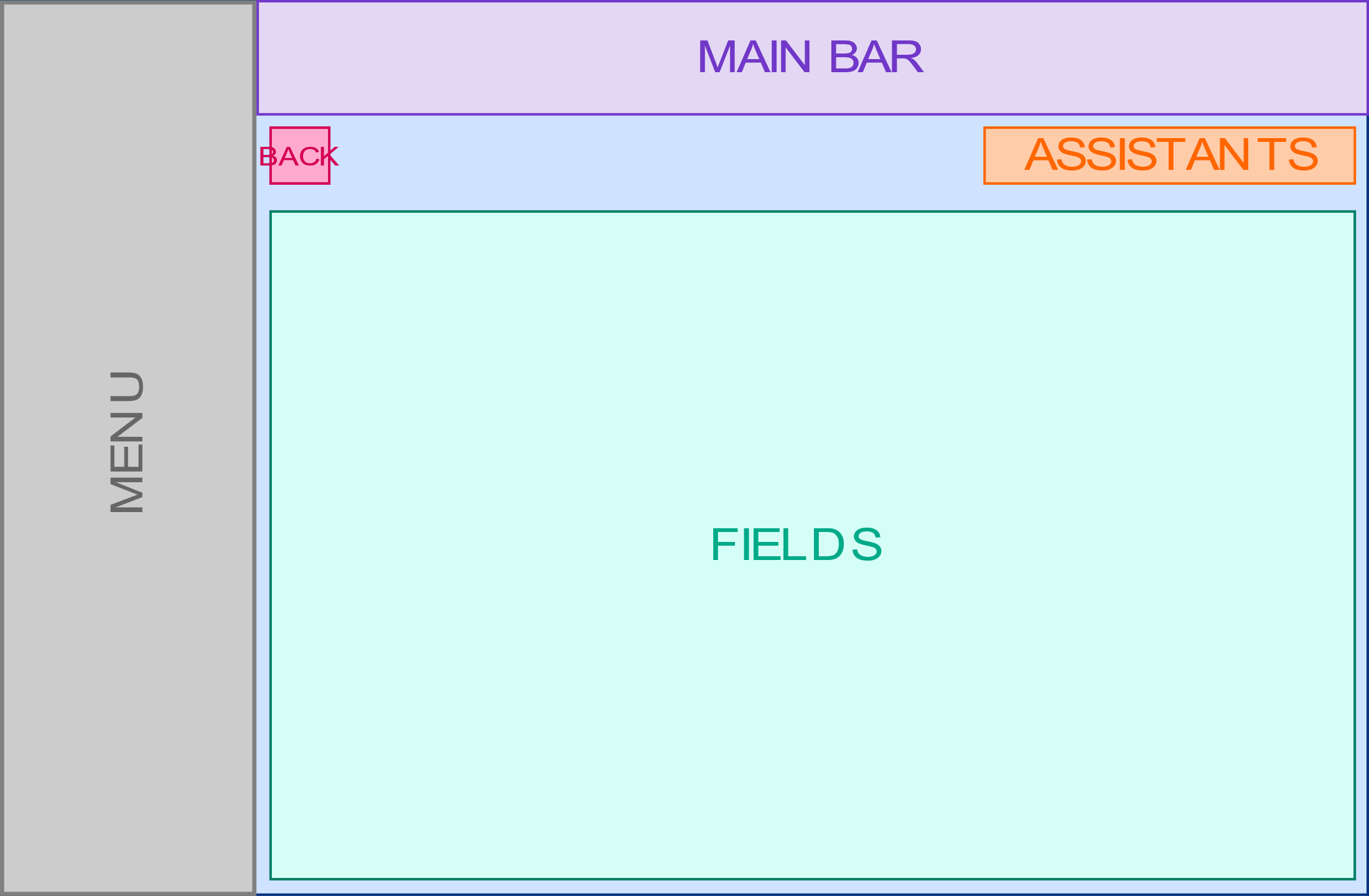
# Screen 3 – Incident File Screen

## Purpose

This screen requests the EVSM API to retrieve the data of one particular incident and display it.

The screen appears in read-only mode and a button allows to switch to edit mode.

## Zoning



### Components

#### Menu

Same as Home Screen

#### Main bar

Same as Home Screen

#### Back

This button allows the user to leave the page and come back to the previous grid, if possible at the same scroll position.

#### Assistants

Display some buttons as shortcuts to frequently used assistants.

A button on the right shows the list of the remaining assistants.

The **assistant should be displayed as a popup**. **At least one example with several pages** (including the usual *"Next"* and *"Close"* buttons) should be implemented.

There must be an *"Edit"* assistant that allows the file to be edited (see Fields).

#### Fields

Displays the fields of the file.

When the page appears, the fields are displayed as read-only. In this mode, no input field or JavaScript component should be loaded or built. These elements will be added dynamically when clicking on the *"Edit"* assistant (see below).

Clicking **the *"Edit"* assistant switches the page to edit mode** and makes all fields editable.

**At least one date field** should be proposed.

# Screen 4 & 5 – Employees List and Employee File Screens

This screen requests the EVSM API to retrieve the list of the employees and display it.

It is the same as the Incidents List and Incident File screens.

# Global requirements

#### Performance indicators

Show the time required for the framework to process the data and display it.

#### Separation of modules

Split the code into several modules to illustrate how the code could be organized by service need using this framework.

#### Apache server

As Apache is the server used in production, the demo project should run on this server.

#### Unit tests

Unit tests should be created to test the sources and demonstrate the ease or difficulty of their implementation.

# Teams

The following teams have been composed :

|  |  |
| --- | --- |
| Aurelia | DELAUNAY Cyriaque  DEMANGE Didier  DOMINGUES Raphael |
| Angular 2/4 | BARHOUNI Younes  EISENKREMER Olivier  GRIMELLI Vincent |
| Ember.js | DANIEL Christophe  KOBANA Constant  MAURER Thomas |