

ДОДАТОК А**КЛАС «MAINVIEW»**

```
@CssImport("styles/main-layout.css")
public class MainView extends AppLayout {

    private final Label titleProjectName = new Label("Soccer Administrator");
    private final Image logo = new Image("images/ball.png", "Login logo");

    private final VerticalLayout headerText = new VerticalLayout();
    private final HorizontalLayout appName = new HorizontalLayout();
    private final HorizontalLayout header = new HorizontalLayout();

    private SearchBar searchBar = new SearchBar();
    private Div searchBarWrapper = new Div(searchBar);

    private Avatar avatar = new Avatar();
    private MenuBar menuBar = new MenuBar();

    private NaviMenu menu;
    private Paragraph viewTitle;
    private SecurityService securityService;

    @Autowired
    public MainView(SecurityService securityService) {
        this.securityService = securityService;
        setPrimarySection(Section.DRAWER);
    }
}
```

```
this.setDrawerOpened(true);

addToNavbar(createHeaderContent());

addToDrawer(createDrawerContent());

getElement().addEventListener("search-focus", e -> {
    getElement().classList.add("hide-navbar");
});

getElement().addEventListener("search-blur", e -> {
    getElement().classList.remove("hide-navbar");
});
}

private Component createHeaderContent() {
    DrawerToggle toggle = new DrawerToggle();
    toggle.addClassName("text-secondary");
    toggle.addThemeVariants(ButtonVariant.LUMO_CONTRAST);
    toggle.getElement().setAttribute("aria-label", "Menu toggle");
    toggle.getStyle().set("color", "#86C232");

    viewTitle = new Paragraph();
    viewTitle.getStyle().set("color", "#fff");
    viewTitle.getStyle().set("font-size", "1.5em");
    viewTitle.getStyle().set("font-weight", "bolder");
    viewTitle.getStyle().set("flex-shrink", "0");
    viewTitle.setClassName("app-title");

    searchBarWrapper.getStyle().set("position", "relative");
```

```

        UserDetails currentUser = (UserDetails)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
        avatar.getElement().getStyle().set("background-color", "mintcream");
        avatar.setName(currentUser.getUsername());
        MenuItem profile = menuBar.addItem(avatar);

        Button logoutLink = new Button("Logout", event ->
securityService.logout());
        logoutLink.addThemeVariants(ButtonVariant.LUMO_TERTIARY);

        profile.getSubMenu().addItem(currentUser.getUsername());
        profile.getSubMenu().addItem(logoutLink);

        header.add(toggle, viewTitle, searchBarWrapper, menuBar);
        header.expand(searchBarWrapper);

        header.getStyle().set("background-color", "#222629");
        header.setAlignItems(FlexComponent.Alignment.CENTER);
        header.setMinWidth("100%");
        header.addClassNames("bg-base", "border-b", "border-contrast-10", "box-
border", "flex", "h-xl", "items-center",
        "w-full");
        return header;
    }

    private void configureSearchBar(SearchBar searchBar) {
        if
(searchBarWrapper.getChildren().collect(Collectors.toList()).contains(this.searchB
ar)) {

```

```
        searchBarWrapper.remove(this.searchBar);  
    }
```

```
        searchBar.setVisible(true);  
        searchBar.getActionButton().setVisible(false);  
        searchBar.setPlaceholder("Search...");  
        searchBar.getElement().getStyle().set("position", "absolute");  
        searchBar.getElement().getStyle().set("top", "-24px");  
        searchBar.getElement().getStyle().set("flex-shrink", "1");  
        searchBar.getElement().setAttribute("class", "main-search-bar");
```

```
        this.searchBar = searchBar;  
        searchBarWrapper.add(searchBar);  
    }
```

```
private String getCurrentPageTitle() {  
    PageTitle title = getContent().getClass().getAnnotation(PageTitle.class);  
    return title == null ? "" : title.value();  
}
```

```
private SearchBar getSearchBar() {  
    return getContent() instanceof WithSearchBar ? ((WithSearchBar)  
getContent()).getSearchBar() : null;  
}
```

```
private Component createDrawerContent() {  
    titleProjectName.getStyle().set("color", "#86C232");  
    titleProjectName.getStyle().set("font-size", "1.5em");  
    titleProjectName.getStyle().set("font-weight", "bolder");
```

```

headerText.add(titleProjectName);
headerText.setWidthFull();
headerText.setMargin(false);
headerText.setPadding(false);
headerText.setSpacing(false);

```

```

appName.add(logo, headerText);
appName.setPadding(true);
appName.setPadding(false);
appName.getStyle().set("margin", "20px 0px 20px 20px");
appName.setAlignItems(FlexComponent.Alignment.END);

```

```

logo.setMaxWidth("50px");
logo.setMaxHeight("50px");

```

```

com.vaadin.flow.component.html.Section section = new
com.vaadin.flow.component.html.Section(appName,
    createNaviMenu());
section.addClassNames("flex", "flex-col", "items-stretch", "max-h-full",
"min-h-full");
section.getStyle().set("overflow", "auto").set("background-
color", "#222629");

```

```

section.setSizeFull();
return section;
}

```

```

@Override
protected void afterNavigation() {
    super.afterNavigation();
}

```

```

viewTitle.setText(getCurrentPageTitle());
menu.getNaviItems().stream().filter(naviItem
naviItem.getSubItems().size() > 0).forEach(naviItem -> {
    naviItem.getSubItems().forEach(subItem -> {
        if (subItem.getNavigationTarget().getAnnotation(Route.class).value()
.equals(Objects.requireNonNull(getLocation()).getPath().substring(1))) {
            naviItem.getExpandCollapse().click();
        }
    });
});

if (getSearchBar() != null) {
    SearchBar childSearchBar = getSearchBar();
    configureSearchBar(childSearchBar);
} else {
    if
(searchBarWrapper.getChildren().collect(Collectors.toList()).contains(this.searchB
ar)) {
        searchBarWrapper.remove(this.searchBar);
    }
}

private URI getLocation() {
    VaadinServletRequest request = (VaadinServletRequest)
VaadinService.getCurrentRequest();
    StringBuffer uriString = request.getRequestURL();
    try {
        return new URI(uriString.toString());
    }
}

```

```

    } catch (URISyntaxException ex) {
        System.out.println("Exception of URL reading" + ex);
        return null;
    }
}

private NaviMenu createNaviMenu() {
    menu = new NaviMenu();

    if (SecurityUtils.isAccessGranted(MyFieldsView.class)) {
        menu.addNavItem(VaadinIcon.MODAL_LIST, "My fields",
MyFieldsView.class);
    }

    if (SecurityUtils.isAccessGranted(FieldOfferView.class)) {
        menu.addNavItem(VaadinIcon.BULLETS, "Fields",
FieldOfferView.class);
    }

    if (SecurityUtils.isAccessGranted(FieldBookView.class)) {
        menu.addNavItem(VaadinIcon.CALENDAR_USER, "Bookings",
FieldBookView.class);
    }

    NaviItem administration = new NaviItem(VaadinIcon.AUTOMATION,
"Administration", null);

    if (SecurityUtils.isAccessGranted(CityView.class)) {
        menu.addNavItem(administration , VaadinIcon.WORKPLACE, "Cities",
CityView.class);
    }
}

```

```

    }

    if (SecurityUtils.isAccessGranted(UserView.class)) {
        menu.addNavItem(administration, VaadinIcon.NURSE, "Users",
UserView.class);
    }

    if (SecurityUtils.isAccessGranted>AboutUsView.class)) {
        menu.addNavItem(VaadinIcon.INFO_CIRCLE_O, "About us",
AboutUsView.class);
    }

    if (SecurityUtils.isAccessGranted>AboutUsView.class)) {
        menu.addNavItem(VaadinIcon.PHONE_LANDLINE, "Contact us",
AboutUsView.class);
    }

    return menu;
}

private static Anchor createLogoutLink(String contextPath) {
    final Anchor a = populateLink(new Anchor(),
VaadinIcon.ARROW_RIGHT, "Logout");
    a.setHref(contextPath + "logout");
    return a;
}

private static <T extends HasComponents> T populateLink(T a, VaadinIcon
icon, String title) {

```

```
HorizontalLayout horizontalLayout = new HorizontalLayout(icon.create(),
new Paragraph(title));
horizontalLayout.setMargin(false);
horizontalLayout.setPadding(false);

horizontalLayout.setJustifyContentMode(FlexComponent.JustifyContentMode.ST
ART);
horizontalLayout.setAlignItems(FlexComponent.Alignment.CENTER);

a.add(horizontalLayout);
return a;
}
}
```