EasyShift Final Report

Final report for CSCI 483. Written by Travis MacDonald and Prahar Ijner.

Introduction

Background

Additional efforts by authors

In addition to concepts covered in the course, we researched and implemented the following to enhance the functionality and usability of EasyShift.

- Database implementation and connection
- Error handling
- Cascading style sheet
- Container management
- UI Components
  - DatePicker
  - Timeline
  - Card
  - Dialog
  - TabMenu
  - Ajax - partial rendering and processing

Website design and workflow

Home page (landing page)

Login page

Registration page

Dashboard-menu

The dashboard menu is the top bar seen by the user upon login. This consists of navigation links for the pages `dashboard.xhtml`, `calendar.xhtml`, `edit_events.xhtml`, and a logout button. The dashboard menu is defined as a template using the `ui:composition` from JSF-Facelets and included into each of the pages the user interacts with after logging in. The dashboard's edit events button is a conditionally rendered element that is displayed only when the logged in user has manager level access.

XHTML source: `project/src/main/webapp/template/dashMenu.xhtml`

Dashboard-menu for managers

Dashboard home

The dashboard home is the first page the user sees after successfully logging into EasyShift. This interface displays the user's current company and the ID of their manager.

Additionally, the dashboard home displays the current availability of the user for each day of the week. The user is has the ability to add or remove availability on any of the days and submit these changes to the system. The interface to facilitate the update of availability uses the components `p:dataTable`, `p:datePicker`, and `p:commandButton`, some of which are conditionally rendered and conditionally editable. The system to edit availability is build with the following features.

- **Add or remove availability on any day** : The user can indicate their abailability on any day of the week by setting hours when they are willing to be scheduled. This can include night shifts, part time abailability, and unabailability on any day(s).

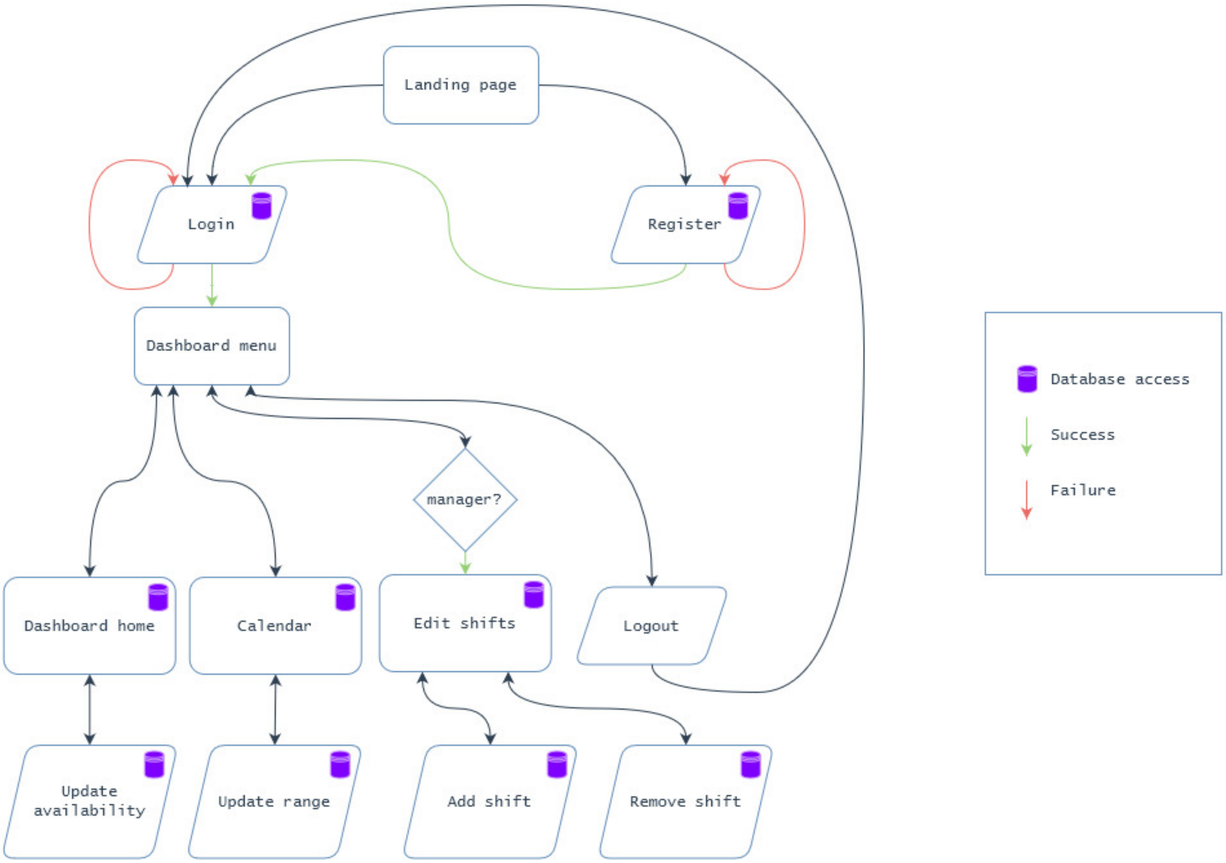

Figure 1: Implemented workflow

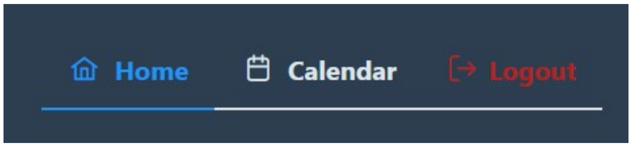

Figure 2: Dashboard-menu for non-managers

- **Add multiple availabilities on a single day** : It is quite common for part-time workers to have multiple abailabilities in a given day, between which they have other engagements. The multiple abailability feature allows them to indicate all these times for easier and more efficient scheduling by the manager.
- **Changes stay local until they are submitted** : In order to avoid un-necessary update events on the database in cases where the user erroneously updated their availability, the changes are not submitted to the database until the user clicks the submit changes button.

XHTML source: `project/src/main/webapp/dashboard.xhtml`

Dashboard calendar

The dashboard calendar shows the availability and shifts scheduled for all the employees in the current user's company. By default, the calendar loads the availabilities and shifts for 7 days from the time of loading the page. The default 7 day range facilitates low resource usage while showing the most immidietely needed information. The user is able to change the range by updating the from and to dates above the timeline that displays the events.

The main componenets of the dashboard interface are `p:timeline`, `p:commandButton`, and `p:datePicker`. It should be noted that the `p:timeline` component uses the server's timezone as the default timezone. While using this with docker containers, this may display the times in UTC (depending on how the image is initialized). In order to avoid confusion, the default time-zone has been set to *America/Anguilla* to correspond to ADT.

XHTML source: `project/src/main/webapp/calendar.xhtml`

Dashboard edit-events

The dashboard edit-events page is exclusively available to users who are managers. This is to ensure that users without appropriate authorization are not able to add or delete shifts. The page can be visited using the *Edit shifts* tab in the dashboard menu. The layout of this page is identical to dashboard calendar with the exception of two additional buttons: *Add shift* and *Remove shift*.

- **Add shift** : Clicking on the *Add shift* button opens up a `p:dialog` with a `p:selectOneMenu` to select the name of the employee that is to be scheduled. The dialog box consists of two `p:datePicker` elements to select the start and end date and time for the shift to be added. Additionally, the scheduling manager can add notes to the shift for the employee. The shift is submitted by clicking the *Add* button at the bottom of the dialog.

- **Remove shift** : Clicking the *Remove shift* button opens another `p:dialog` with a `p:dataTable` that consists of all the shifts within the *From* and *To* range on the edit shifts page. The user can delete shift(s) by clicking the delete button on the row corresponding to the shift.

XHTML source: `project/src/main/webapp/edit_shifts.xhtml`

Logout

The logout tab in the dashboard menu allows the user to log out of their account. Since all the user's activity after logging in is scoped within a session, the logout is performed by ending the session and re-directing the user to the login page.

Styling

To ensure consistent design patterns and colors throughout the web pages, we defined a style sheet (see `project/src/main/webapp/styles.css`). This tool significantly aided us in improving the visual appeal of the application and setting parameters to distinguish calendar events, i.e., available and scheduled events.

Database

A database is a critical part of any web application. For EasyShift, we use a database to store all information about employees, companies, and shifts. The database server used is MySQL 8.0. The database schema used is defined in the figure below. It should be noted that the schema contains the table `exception_times` which is a feature not used in the current implementation, but is intended to indicate vacation days or sick days for an employee or a company.

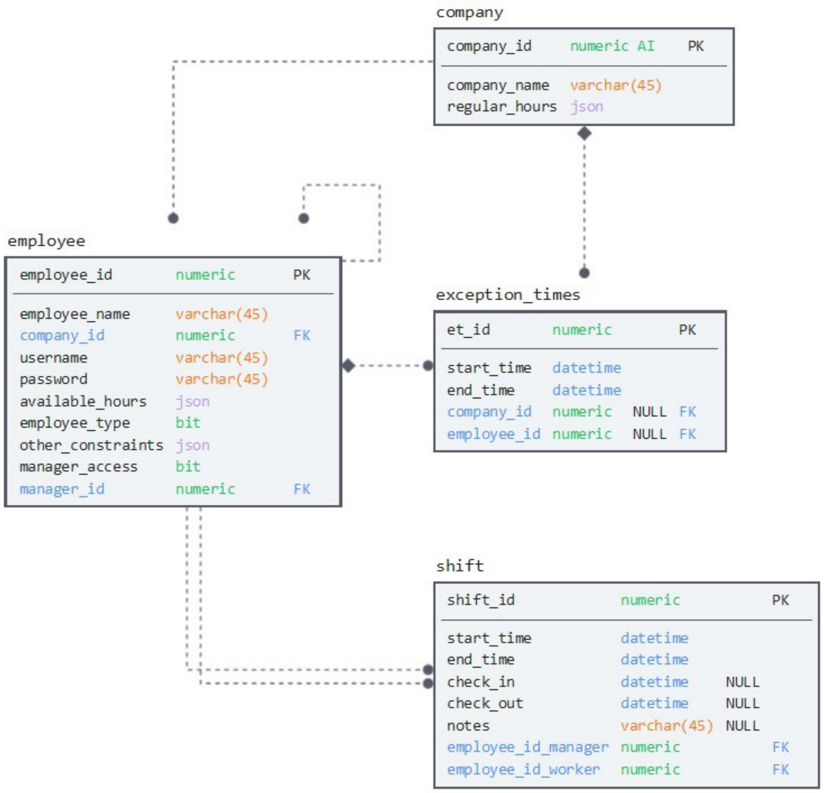

Figure 3: Database schema

Future steps