CS 319-1 Group 8

Deliverable 1 - Final

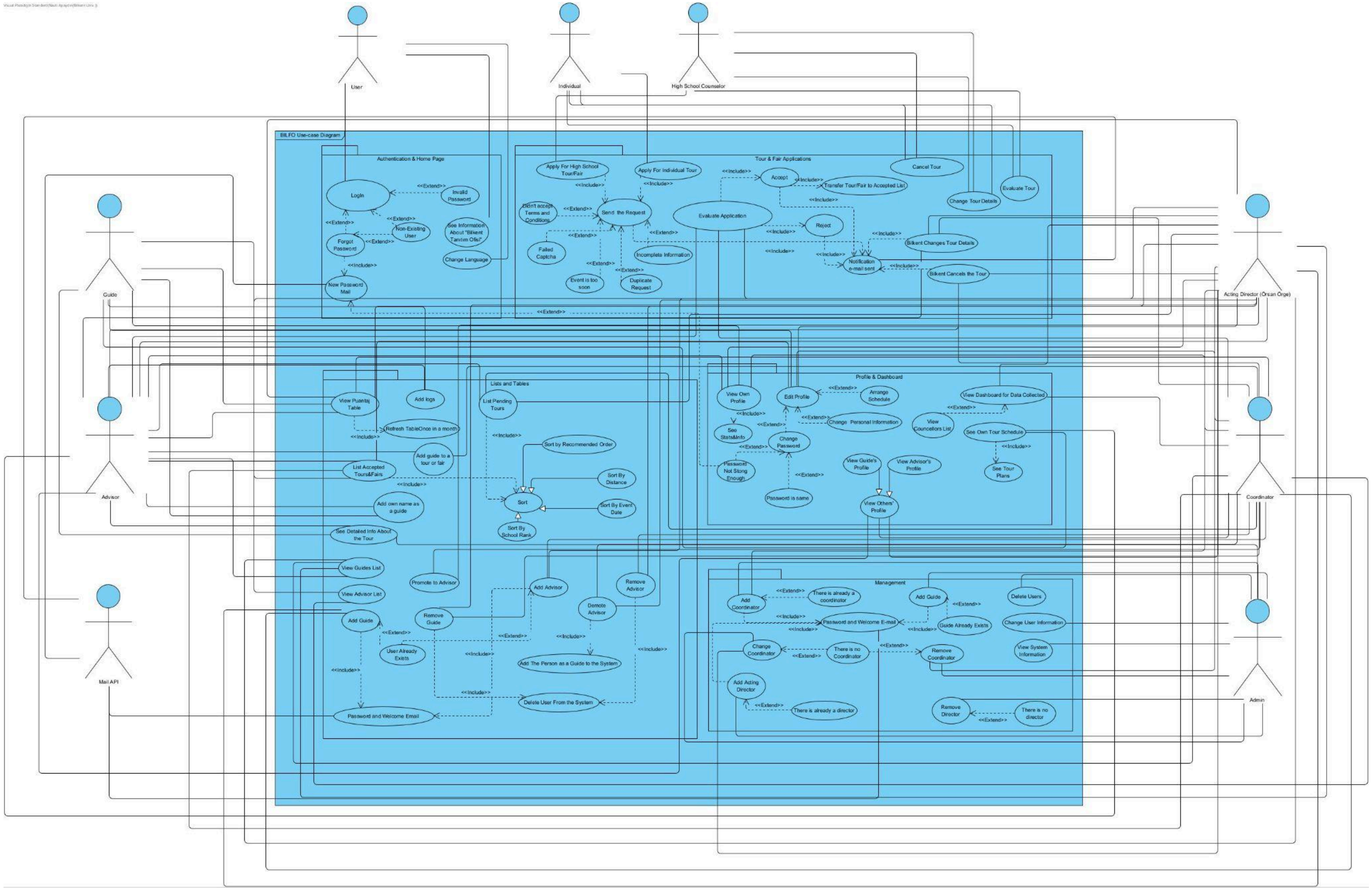
Bilkent Tanıtım Ofisi Management System

Short-name: BILFO

12.11.2024

Ata Uzay Kuzey	22203050
Duru Solakoğlu	22102720
Emir Görgülü	22202834
Kerem Demirören	22203380
Nazlı Apaydın	22202104

1) Use Case Diagram



2) Use Case Textual Descriptions

Authentication & Home Page Package

1. **Name:** Change Language
2. **Participating actor:** User
3. **Entry condition:** User clicks the button to change the language.
4. **Exit condition:** The language is changed.
5. **Flow of events:**
   - **5.1.** User clicks the button to change the language.
   - **5.2.** The user's website changes its language. If the language is Turkish, it changes to English. If the language is English, it changes to Turkish.
6. **Special /quality requirements:** This preference should be stored locally on the user's computer.

1. **Name:** See Information About "Bilkent Tanıtım Ofisi"
2. **Participating actor:** User
3. **Entry condition:** User clicks the button to see information.
4. **Exit condition:** Information about the “Bilkent Tanıtım Ofisi” is shown.
5. **Flow of events:**
   - **5.1.** User clicks the button to see information.
   - **5.2.** Information about the “Bilkent Tanıtım Ofisi” is shown. This includes an “About Us” page or a guide on how to apply for a tour.

1. **Name:** Log in
2. **Participating actor:** User
3. **Entry condition:** User clicks the button to log in.
4. **Exit condition:** If authentication succeeds, the user logs in, if not, an error is shown.
5. **Flow of events:**
   - **5.1.** User clicks the button to log in.
   - **5.2.** User enters user id and password.
   - **5.3.** If the user id doesn’t exist in the system, or password doesn’t match with the provided id, an error is shown. Otherwise, the user logs in to the site.
6. **Special /quality requirements:** This process can only be completed 5 times at a time for security measures.

1. **Name:** Invalid Password
2. **Participating actor:** User
3. **Entry condition:** User tries to log in but the password doesn’t match with the stored password of the user id.
4. **Exit condition:** An error is shown telling the user about the situation.
5. **Flow of events:**

    5.1. User is shown an error message for entering the wrong password.
    5.2. User goes back to the login page to try again.

1. **Name:** Non-Existing User
2. **Participating actor:** User
3. **Entry condition:** User tries to log in but the user id doesn’t exist in the database.
4. **Exit condition:** An error is shown telling the user about the situation.
5. **Flow of events:**
    5.1. User is shown an error message for entering an invalid user id.
    5.2. User goes back to the login page to try again.

1. **Name:** Forgot Password
2. **Participating actor:** User
3. **Entry condition:** User clicks the forgot password button on the login page.
4. **Exit condition:** If a user enters an invalid user id, an error is displayed. If the id is valid, an email to change the user's password is sent to the user's mail.
5. **Flow of events:**
    5.1. User clicks the forgot password button on the login page.
    5.2. User enters their id.
    5.3. If the id is valid, an email is sent. If the id is invalid, an error message is shown to the user.

1. **Name:** New Password Mail
2. **Participating actor:** Mail API
3. **Entry condition:** User clicks “Forgot Password”.
4. **Exit condition:** The password of the user is changed.
5. **Flow of events:**
    5.1. User clicks the link sent to them through the mail.
    5.2. User enters their new password. If the password isn’t strong enough, this step is repeated.
    5.3. The password of the user is changed to the new password.
6. **Special /quality requirements:** This process should be completed within 2 minutes, else the link will expire for security measures.

Management Package

1. **Name:** Add Coordinator
2. **Participating actor:** Admin or Director
3. **Entry condition:** Admin or Director clicks add coordinator button.
4. **Exit condition:** A new coordinator is added to the system.
5. **Flow of events:**
    5.1. Admin or Director clicks on the add coordinator button.

5.2. The system checks if a coordinator exists, if it does, this continues to "There is already a Coordinator".
5.3. The user is directed to the fill coordinator information page.
5.4. User enters the info of the new coordinator.
5.5. New coordinator is added to the system.
5.6. Success message is shown to the user.

1. **Name:** Password and Welcome Email
2. **Participating actor:** Mail API
3. **Entry condition:** A new Coordinator or Director is created.
4. **Exit condition:** An email is sent to the new Coordinator or Director.
5. **Flow of events:**
   5.1. A new Coordinator or Director is added into the system.
   5.2. An email notifying the new Coordinator or Director of their addition and their password is sent.
6. **Special / quality requirements:** The process should take less than two minutes to speed up the registration.

1. **Name:** There is already a coordinator.
2. **Participating actor:** Admin or Director
3. **Entry condition:** Admin or Director clicked the add coordinator button but there is already a coordinator in the system.
4. **Exit condition:** An error message is shown to the admin.
5. **Flow of events:**
   5.1. Admin or Director clicks on the add coordinator button.
   5.2. System discovers there is already a coordinator on the server.
   5.3. An error message informing the Admin or the Director is sent.

1. **Name:** Change Coordinator
2. **Participating actor:** Admin or Director
3. **Entry condition:** Admin or Director clicks the change coordinator button.
4. **Exit condition:** Current coordinator is demoted and the chosen advisor is promoted to coordinator.
5. **Flow of events:**
   5.1. Admin or Director clicks on the change coordinator button.
   5.2. If there isn't already a coordinator, this will continue into "There is no Coordinator".
   5.3. The Admin or the Director chooses an advisor to promote to coordinator.
   5.4. The current Coordinator is demoted to being an advisor.
   5.5. The chosen advisor is promoted to a coordinator.

1. **Name:** There is no Coordinator
2. **Participating actor:** Admin or Director

3. **Entry condition:** Admin or Director tries to do something with the Coordinator, but they don't exist in the system.
4. **Exit condition:** An error message is shown.
5. **Flow of events:**
   - **5.1.** Admin or Director tries to either remove or change the current coordinator.
   - **5.2.** The system recognizes there is no coordinator.
   - **5.3.** An error message is sent.

1. **Name:** Remove Coordinator
2. **Participating actor:** Admin or Director
3. **Entry condition:** Admin or Director clicks to the Remove Coordinator button.
4. **Exit condition:** The coordinator is removed.
5. **Flow of events:**
   - **5.1.** Admin or Director clicks to the Remove Coordinator button.
   - **5.2.** The system checks if a coordinator exists, if not, this continues to "There is no Coordinator".
   - **5.3.** The coordinator is removed from the system.

1. **Name:** Add Acting Director
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks to the Add Director button.
4. **Exit condition:** A new director is added to the system.
5. **Flow of events:**
   - **5.1.** Admin clicks to the Add Coordinator button.
   - **5.2.** The system checks if an Acting Director exists, if it does, this continues to "There is already an Acting Director".
   - **5.3.** The admin is directed to the fill director information page.
   - **5.4.** Admin enters the info of the new director.
   - **5.5.** The director is added to the system.
   - **5.6.** Success message is shown to the admin.

1. **Name:** There is Already a Director
2. **Participating actor:** Admin
3. **Entry condition:** Admin tries to add a Director but there is already a Director.
4. **Exit condition:** An error message is shown to the admin.
5. **Flow of events:**
   - **5.1.** Admin tries to add a director to the system.
   - **5.2.** There is already a director in the system.
   - **5.3.** An error message is shown.

1. **Name:** Remove Director
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks to the Remove Director button.

4. **Exit condition:** The director is removed.
5. **Flow of events:**
    - **5.1.** Admin clicks to the Remove Director button.
    - **5.2.** The system checks if a director exists, if not, this continues to “There is no Director”.
    - **5.3.** The director is removed from the system.

1. **Name:** There is no Director
2. **Participating actor:** Admin
3. **Entry condition:** Admin tries to do something with the Director, but they don’t exist in the system.
4. **Exit condition:** An error message is shown.
5. **Flow of events:**
    - **5.1.** Admin tries to either remove or change the current director.
    - **5.2.** The system recognizes there is no director.
    - **5.3.** An error message is sent.

1. **Name:** Change User Information
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks the button for changing information of a user in the managing users list.
4. **Exit condition:** The information of the user is changed.
5. **Flow of events:**
    - **5.1.** Admin clicks the button to change the information of a user.
    - **5.2.** Admin enters new information.
    - **5.3.** The information of the user is changed to the new information.

1. **Name:** Delete Users
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks the button for deleting the user in the managing users list.
4. **Exit condition:** The user is deleted.
5. **Flow of events:**
    - **5.1.** Admin clicks the button to delete the user.
    - **5.2.** The user is deleted.

1. **Name:** Add Guide
2. **Participating actor:** Admin, Director
3. **Entry condition:** User clicks to the add guide button.
4. **Exit condition:** A new guide is added to the system
5. **Flow of events:**
    - **5.1.** Admin clicks the button to add a guide.
    - **5.2.** The user is directed to the fill guide information page.
    - **5.3.** User enters the info of the new guide.

   5.4. New guide is added to the system.
   5.5. Success message is shown to the user.
   5.6. Welcome email is sent to the new guide.

1. **Name:** View System Information
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks the button viewing the system information.
4. **Exit condition:** The admin sees the system information.
5. **Flow of events:**
   5.1. Admin clicks the button to view the system information.
   5.2. Admin sees the system information.

List and Tables Package

1. **Name**: View Puantaj Table
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry condition:** User clicks to the view puantaj table button.
4. **Exit Condition:** The puantaj table is shown to the user.
5. **Flow of events:**
   5.1. The user clicks to the view puantaj table button.
   5.2. The puantaj table is shown with the logs pulled from the database.

1. **Name:** Add Logs
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Enter condition:** User clicks to the Add Log button.
4. **Exit Condition:** The log is added to the database.
5. **Flow of events:**
   5.1. The user clicks to the Add Log button.
   5.2. User selects the day and hours to add his/her log.
   5.3. The log is added to the database.
   5.4. A success message is sent.
   5.5. User is returned to the puantaj table.

1. **Name:** Refresh Table Once in a Month
2. **Entry condition:** The time for refresh becomes zero.
3. **Exit Condition:** The puantaj table is refreshed.
4. **Flow of events:**
   4.1. The timer which is for refreshing action becomes zero.
   4.2. The puantaj table is refreshed.
5. **Special / quality requirements:** The refresh process should complete in less than 5 seconds. Data should be backed up before refreshing to avoid accidental data loss.

1. **Name:** Sort

2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry condition:** The dropdown menu for sorting above the table is pressed.
4. **Exit Condition:** One of the sorting options is chosen.
5. **Flow of events:**
   - **5.1.** The sorting menu is pressed.
   - **5.2.** One of the sorting options is chosen.
6. **Special / quality requirements:** Sorting should take no more than 2 seconds for tables. Consistency should be ensured in sorting criteria (e.g. YKS rank, date format) to prevent misordering.

1. **Name:** Sort by School Rank
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry condition:** The Sort by School Rank option from the dropdown menu is chosen.
4. **Exit Condition:**The list is sorted according to schools' ranks which is determined by YKS results and Bilkent legacy..
5. **Flow of events:**
   - **5.1.** The Sort by School Rank option is selected.
   - **5.2.** The tours are sorted according to the coming schools' ranks.
   - **5.3.** From the highest ranked school to lowest, the list is shown to the user.

1. **Name:** Sort by Event Date
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry condition:** The Sort by Event Date option from the dropdown menu is chosen.
4. **Exit Condition:** The list is sorted according to the event date.
5. **Flow of events:**
   - **5.1.** The Sort by Event Date option is selected.
   - **5.2.** The tours are sorted according to the coming tours' or fairs' date.
   - **5.3.** Events are shown from the most recent to the least.

1. **Name:** Sort by Distance
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry condition:** The Sort by Distance option from the dropdown menu is chosen.
4. **Exit Condition:** The list is sorted according to the distance from Bilkent.
5. **Flow of events:**
   - **5.1.** The Sort by Distance option is selected.
   - **5.2.** The tours are sorted according to the distance from Bilkent.
   - **5.3.** From the furthest to the nearest, the list is shown to the user.

1. **Name:** Sort by Recommended Order
2. **Participating actor:** Guide, Advisor, Coordinator or Director

3. **Entry condition:** The Sort by Recommended Order option from the dropdown menu is chosen
4. **Exit Condition:** The list is sorted according to our algorithm regarding the High School category, city they are coming from, and lastly the first come first serve principle.
5. **Flow of events:**
   - **5.1.** The Sort by Recommended Order option is selected.
   - **5.2.** The tours are sorted according to our algorithm we will implement.
   - **5.3.** The list is shown to the user in a priority order.

1. **Name:** List Accepted Tours and Fairs
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry Condition:** User clicks to the List of Accepted Tours and Fairs button.
4. **Exit Condition:** The list of accepted tours and fairs is shown to the user.
5. **Flow of events:**
   - **5.1.** User clicks to the List of Accepted Tours and Fairs button.
   - **5.2.** The accepted tours and fairs are pulled from the database.
   - **5.3.** The list of accepted tours and fairs is shown to the user.
6. **Special / quality requirements:** Loading time should not exceed 2 seconds. Data retrieved should be updated in real-time to reflect the latest statuses.

1. **Name:** List Pending Tours
2. **Participating actor:** Advisor, Coordinator or Director
3. **Entry Condition:** User clicks to the List of Pending Tours and button.
4. **Exit Condition:** The list of pending tours is shown to the user.
5. **Flow of events:**
   - **5.1.** User clicks to the List of Pending Tours and Fairs button.
   - **5.2.** The pending tours are pulled from the database.
   - **5.3.** The list of pending tours is shown to the user.
6. **Special / quality requirements:** Loading time should not exceed 2 seconds. Data retrieved should be updated in real-time to reflect the latest statuses.

1. **Name:** See Detailed Info About the Tour
2. **Participating actor:** Guide, Advisor, Coordinator or Director
3. **Entry Condition:** A tour from the list of accepted tours is selected.
4. **Exit Condition:** The detailed information about the selected tour is shown to the user
5. **Flow of events:**
   - **5.1.** User selects a tour from the accepted tours list.
   - **5.2.** The detailed information such as the participant number, contact information, high school category, time, city etc. is shown about the tour.

1. **Name:** Add Own Name as a Guide

2. **Participating actor:** Guide, Advisor
3. **Entry Condition:** From the accepted tours and fairs the user clicks on the "Guide The Tour" button near the preferred tour or fair which is still unassigned to its necessary number of guides.
4. **Exit Condition:** The name of the user is assigned for the tour or fair and it is stored in the database.
5. **Flow of events:**
   - **5.1.** User clicks the "Guide The Tour" button near the desired unassigned tour or fair.
   - **5.2.** The guide or advisor is assigned to the tour or fair.
   - **5.3.** The information is sent to the database.
   - **5.4.** A success message is sent.
   - **5.5.** The assigned tour is transferred to the guide's or advisor's schedule.
   - **5.6.** User is returned to the accepted tours and fairs list.

1. **Name:** Add Guide to a Tour or Fair
2. **Participating actor**: Advisor, Coordinator
3. **Entry Condition:** From the accepted tours and fairs, advisor or coordinator clicks to the Add Guide button near the desired tour or fair which is still unassigned.
4. **Exit Condition:** The guide is assigned for the tour and it is stored in the database.
5. **Flow of events:**
   - **5.1.** Advisor or coordinator clicks the Add Guide button near the desired unassigned tour or fair.
   - **5.2.** The guide is assigned to the tour if they are available.
   - **5.3.** The information is sent to the database.
   - **5.4.** The assigned tour is transferred to the guide's schedule.
   - **5.5.** User is returned to the accepted tours list.

1. **Name:** Promote to Advisor
2. **Participating actor:** Acting Director, Coordinator
3. **Entry Condition:** Acting director or coordinator clicks to the Promote to Advisor button near the desired guide in the guides list.
4. **Exit Condition:** The former guide is added to the advisors in the database
5. **Flow of events:**
   - **5.1.** Acting Director or coordinator click to the Promote to Advisor button.
   - **5.2.** Selected guide is deleted from the guides in the database.
   - **5.3.** Former guide is added to the advisors in the database.
   - **5.4.** A success message is sent.
   - **5.5.** The updated guides list is shown.

1. **Name:** Add Guide
2. **Participating actor:** Director, Coordinator

3. **Entry Condition:** Acting director or coordinator clicks to the Add Guide button that is near the desired guide.
4. **Exit Condition:** New guide is added to the system.
5. **Flow of events:**
   - **5.1.** Acting Director or coordinator click to the Add Guide button.
   - **5.2.** The user is directed to the fill guide information page.
   - **5.3.** User enters the info of the new guide.
   - **5.4.** New guide is added to the system.
   - **5.5.** Success message is shown to the user.
   - **5.6.** Welcome email is sent to the new guide.

1. **Name:** Password and Welcome Mail is Sent
2. **Participating actor:** Mail API
3. **Entry Condition:** The data is stored in the database.
4. **Exit Condition:** Password and welcome mail is sent to the new user.
5. **Flow of events:**
   - **5.1.** The data is stored in the database.
   - **5.2.** The password and welcome mail is sent.
6. **Special / quality requirements:** The process should take less than one minute to speed up the registration.

1. **Name:** Remove Guide
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** Acting director or coordinator clicks to the Remove Guide button that is near the desired guide.
4. **Exit Condition:** The data of the removed guide is sent to the database.
5. **Flow of events:**
   - **5.1.** Acting Director or coordinator click to the Remove Guide button.
   - **5.2.** Warning message shows up if the user proceeds the flow continues, otherwise nothing happens..
   - **5.3.** After accepting to delete the data is sent to the database.

1. **Name:** Delete User From the System
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The data of the guide or advisor who is going to be removed is sent to the database.
4. **Exit Condition:** The data of the removed guide or advisor is erased from the database.
5. **Flow of events:**
   - **5.1.** The data of the selected guide or advisor is sent to the database.
   - **5.2.** Warning message shows up if the user proceeds and the flow continues, otherwise nothing happens..
   - **5.3.** The guide or advisor is removed from the database.

1. **Name:** View Advisor List
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The user clicks to the View Advisor List button.
4. **Exit Condition:** All advisors in the system are shown through a list.
5. **Flow of events:**
    - **5.1.** The user clicks to the View Advisor List button.
    - **5.2.** Advisors are obtained from the database.
    - **5.3.** Advisors List is shown.

1. **Name:** Add Advisor
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The user clicks to the Add Advisor button.
4. **Exit Condition:** Advisor is stored in the database.
5. **Flow of events:**
    - **5.1.** The user clicks to the Add Advisor button.
    - **5.2.** The user enters the information of advisor
    - **5.3.** Advisor data is sent to the database.
    - **5.4.** Data is stored in the database.
    - **5.5.** Success message is shown to the user.

1. **Name:** Demote Advisor
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The user clicks to the Demote Advisor button near the desired advisor.
4. **Exit Condition:** Advisor is demoted and the database is updated.
5. **Flow of events:**
    - **5.1.** The user clicks to the Demote Advisor button.
    - **5.2.** Warning message is shown and if the user proceeds the flow continues, otherwise nothing happens.
    - **5.3.** Advisor is removed from the advisor list in the database.
    - **5.4.** Former advisor is stored in the guides list in the database.
    - **5.5.** Success message is shown to the user.

1. **Name:** Remove Advisor
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The user clicks to the Remote Advisor button near the desired advisor.
4. **Exit Condition:** Advisor is removed and the database is updated.
5. **Flow of events:**
    - **5.1.** The user clicks to the Remove Advisor button.
    - **5.2.** Warning message is shown and if the user proceeds the flow continues, otherwise nothing happens.
    - **5.3.** Advisor is removed from the advisor list in the database.
    - **5.4.** Success message is shown to the user.

Tour & Fair Applications Package

1. **Name:** Evaluation of the Applications
2. **Participating actor:** Advisor, Acting Director, Coordinator
3. **Entry condition:** Advisor, Coordinator or Acting Director opens the pending requests list.
4. **Exit condition:** Evaluation of an application is completed, either accepted or rejected.
5. **Flow of events:**
    - **5.1.** Pending Requests list is updated after every request.
    - **5.2.** Advisor, Acting Director or Coordinator can check the pending requests list by clicking the Pending Requests button.
    - **5.3.** The selected request is evaluated considering the order when sorted.
    - **5.4.** The request will either be accepted or rejected.
    - **5.5.** If accepted, the request is transferred to the Accepted Tour/Fair list and continues to “Notification Email Sent”.
    - **5.6.** If rejected the process continues with “Notification Email Sent”.

1. **Name:** Notification Email Sent
2. **Participating actor:** Mail API
3. **Entry condition:** The process requires a notification email to inform the individual or counselor.
4. **Exit condition:** E-mail is sent.
5. **Flow of events:**
    - **5.1.** A process requires a notification to inform the individual or counselor.
    - **5.2.** An email will be sent to the user.
6. **Special / quality requirements:** For each type of notification email (acceptance/rejection) the contents of the mail will be different. For example after sending a request there will be a link for change information or canceling the event. But when Bilkent cancels a tour there won”t be any link in the mail.

1. **Name:** Bilkent Cancels the Tour
2. **Participating actor:** Advisor, Acting Director, Coordinator
3. **Entry condition:** Advisor, Coordinator or Acting Director opens the Accepted Tours list and clicks the cancel on a selected tour.
4. **Exit condition:** Tour is canceled and database is updated.
5. **Flow of events:**
    - **5.1.** Advisor, Coordinator or Acting Director opens the Accepted Tours list.
    - **5.2.** Advisor, Coordinator or Acting Director selects the tour and clicks the cancel button.
    - **5.3.** Email to inform of the cancellation is sent.

1. **Name:** Bilkent Changes Tour Details
2. **Participating actor:** Advisor, Acting Director, Coordinator

3. **Entry condition:** Advisor, Coordinator or Acting Director opens the Accepted Tours list and clicks the edit button on a selected tour.
4. **Exit condition:** Clicks the save button.
5. **Flow of events:**
   - **5.1.** Advisor, Coordinator or Acting Director opens the Accepted Tours list.
   - **5.2.** Advisor, Coordinator or Acting Director selects the tour and clicks the edit button.
   - **5.3.** Advisor, Coordinator or Acting Director changes the schedule of the tour.
   - **5.4.** Email about the updated information is sent to the user.

1. **Name:** Evaluate Tour
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** The tour ends.
4. **Exit condition:** User fills the feedback form.
5. **Flow of events:**
   - **5.1.** After a tour ends, an email is sent to the user.
   - **5.2.** E-mail includes a link that directs the user to the website to get the user’s feedback.
   - **5.3.** The user rates their experience and tour guides.

1. **Name:** Apply For Individual Tour
2. **Participating actor:**
3. **Entry condition:** Individual enters the website and opens the application form button.
4. **Exit condition:** Individual completes the application, either successfully or unsuccessfully.
5. **Flow of events:**
   - **5.1.** The Individual opens the website and opens the application form.
   - **5.2.** The Individual selects the application type, this continues on the application form
   - **5.3.** The Individual fills out the necessary information.
   - **5.4.** The Individual reads and accepts the Terms and Conditions or else the application will fail.
   - **5.5.** The Individual does the Captcha, if the user fails, it will be recognized as a bot and the application will be rejected.
   - **5.6.** If the requested date is too soon, the application will be rejected.
   - **5.7.** If there is incomplete information, the application will be rejected.
   - **5.8.** Otherwise, the application will be successful and sent to the Pending Requests list.
   - **5.9.** Notification mail will be sent to the applicant.
6. **Special / quality requirements:** Rejection at any step ensures that only valid, complete, and non-duplicate applications reach the Pending Requests list, minimizing redundant data and streamlining the review process. This

approach helps maintain the overall quality of applications and reduces system load, ensuring resources are available for eligible applicants.

1. **Name:** Apply For High School Tour/Fair
2. **Participating actor:** High School Counselor
3. **Entry condition:** High School Counselor enters the website and selects the application form.
4. **Exit condition:** High School Counselor completes the application, either successfully or unsuccessfully.
5. **Flow of events:**
    - **5.1.** The High School Counselor opens the website and opens the application form.
    - **5.2.** The High School Counselor selects the application type, this continues on the application form.
    - **5.3.** The High School Counselor fills out the necessary information.
    - **5.4.** The High School Counselor reads and accepts the Terms and Conditions or else the application will fail.
    - **5.5.** The High School Counselor does the Captcha, if the High School Counselor fails, it will be recognized as a bot and the application will be rejected.
    - **5.6.** If the requested date is too soon, the application will be rejected.
    - **5.7.** If there is incomplete information, the application will be rejected.
    - **5.8.** If the high school has already sent a request, the duplicate application will be rejected.
    - **5.9.** Otherwise, the application will be successful and sent to the Pending Requests list.
    - **5.10.** Notification mail will be sent to the counselor.
6. **Special / quality requirements:** Rejection at any step ensures that only valid, complete, and non-duplicate applications reach the Pending Requests list, minimizing redundant data and streamlining the review process. This approach helps maintain the overall quality of applications and reduces system load, ensuring resources are available for eligible applicants.

1. **Name:** Didn’t Accept Terms and Conditions
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** User does not click the terms and conditions checkbox.
4. **Exit condition:** Application will be rejected.
5. **Flow of events:**
    - **5.1.** The user tries to send their application.
    - **5.2.** The system recognizes the checkbox is not selected.
    - **5.3.** An error message is sent.
    - **5.4.** Application will be rejected.

1. **Name:** Failed Captcha
2. **Participating actor:** Individual, High School Counselor

3. **Entry condition:** User fails the captcha before sending the application form.
4. **Exit condition:** Application will be rejected.
5. **Flow of events:**
    - **5.1.** User tries to do the captcha.
    - **5.2.** The bot recognizes the failed pattern of captcha.
    - **5.3.** An error message is sent.
    - **5.4.** Application will be rejected.

1. **Name:** Event is too Soon
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** User enters a date for the event that can not be requested.
4. **Exit condition:** Application will be rejected.
5. **Flow of events:**
    - **5.1.** User enters a date that is too soon for the application.
    - **5.2.** The system recognizes the date is not acceptable.
    - **5.3.** An error message is sent.
    - **5.4.** Application will be rejected.

1. **Name:** Duplicate Request
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** There is another event request by the same individual or high school.
4. **Exit condition:** Application will be rejected.
5. **Flow of events:**
    - **5.1.** User tries to send a request.
    - **5.2.** The system recognizes there is another request by the same high school or individual.
    - **5.3.** An error message is sent.
    - **5.4.** Application will be rejected.

1. **Name:** Incomplete Information
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** User enters incomplete application form.
4. **Exit condition:** Application will be rejected.
5. **Flow of events:**
    - **5.1.** Admin tries to either remove or change the current director.
    - **5.2.** The system recognizes the form is incomplete.
    - **5.3.** An error message is sent.
    - **5.4.** Application will be rejected.

1. **Name:** Cancel Tour
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** User goes to the link provided for the tour and clicks on cancel tour.

4. **Exit condition:** Tour is deleted from the system.
5. **Flow of events:**
    - **5.1.** The user clicks to the link in the mail for the specific tour.
    - **5.2.** In the tour page the user clicks to the cancel tour button.
    - **5.3.** A confirmation pop-up shows and if the user continues the tour will be deleted from the system otherwise nothing happens.

1. **Name:** Change Tour Details
2. **Participating actor:** Individual, High School Counselor
3. **Entry condition:** User goes to the link provided for the tour and clicks on change details.
4. **Exit condition:** The information of the tour at the system will be updated.
5. **Flow of events:**
    - **5.1.** The user clicks to the link in the mail for the specific tour.
    - **5.2.** In the tour page the user clicks to the change details button.
    - **5.3.** The user is directed to the update tour information page.
    - **5.4.** User clicks the update button.
    - **5.5.** Warning message is shown. If the user continues the tour information will be updated in the system otherwise the user will still be in the information page and nothing happens.

Profile & Dashboard Package

1. **Name:** View Own Profile
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “View Own Profile” button.
4. **Exit condition:** Profile information is shown.
5. **Flow of events:**
    - **5.1.** Following the click of the button, a page is shown that includes information about the one’s profile.

1. **Name:** See Stats & Info
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “See Stats & Info” button.
4. **Exit condition:** Tour stats and tour info are shown.
5. **Flow of events:**
    - **5.1.** Following the click of the button, if the profile owner is not an acting director, a page appears that shows stats information about tours.
    - **5.2.** Following the click of the button, if the profile owner is an acting director, a message appears that states “This service is only accessible by Advisors, Coordinators and Guides”.

1. **Name:** Edit Profile
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “Edit Profile” button.
4. **Exit condition:** Advisor, Coordinator, Guide or Acting Director click to “Save Changes” or “Exit Without Saving” button.
5. **Flow of events:**
    - **5.1.** Following the click of the “Edit Profile” button, a page is shown in which changes can be done.
    - **5.2.** If desired, necessary changes are made or the profile remains unchanged.
    - **5.3.** If a change is made, it is updated to the database system.

1. **Name:** Arrange Schedule
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “Arrange Schedule” button.
4. **Exit condition:** Advisor, Coordinator, Guide or Acting Director click to “Save Changes” or “Exit Without Saving” button.
5. **Flow of events:**
    - **5.1.** Following the click of the “Arrange Schedule” button, the profile owner’s weekly schedule is shown.
    - **5.2.** Desired areas in the schedule can be marked as “Available”, “Occupied” or “In Tour”.
    - **5.3.** If a change is made, it is saved to the database by clicking to the “Save Changes” button.

1. **Name:** Change Personal Information
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “Change Personal Information” button.
4. **Exit condition:** Advisor, Coordinator, Guide or Acting Director click to “Save Changes” or “Exit Without Saving” button.
5. **Flow of events:**
    - **5.1.** Following the click of the “Change Personal Information” button, the profile owner’s personal information is shown.
    - **5.2.** Desired personal information can be changed.
    - **5.3.** If a change is made, it is saved to the database by the click to the “Save Changes” button.

1. **Name:** Change Password
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director click to “Change Password” button.
4. **Exit condition:** Password is changed successfully or an error occurred.

5. **Flow of events:**
    - **5.1.** System asks the profile owner to enter the new password.
    - **5.2.** If password requirements meet the standards and are different from the existing one, the new password is saved to the database and a success message is shown.
    - **5.3.** If password requirements do not meet the standards or it is the same with the existing one, an error message is shown.

1. **Name:** Password Not Strong Enough
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry condition:** Advisor, Coordinator, Guide or Acting Director are directed into the page.
4. **Exit condition:** Password is changed successfully or an error occurred.
5. **Flow of events:**
    - **5.1.** System shows the error message “Password Not Strong Enough”.
    - **5.2.** System shows the standards of a password and asks the profile owner to retry.
    - **5.3.** If password requirements meet the standards and are different from the existing one, the new password is saved to the database and a success message is shown.
    - **5.4.** If password requirements do not meet the standards or same with the existing one, a regarding error message is shown.
6. **Special /quality requirements:** Password should be longer than 8 characters and it should include numbers, special characters and capital letters as well as small case letters.

1. **Name:** Password Is Same
2. **Participating actor:** Advisor, Acting Director, Coordinator, Guide
3. **Entry conditions:** Advisor, Coordinator, Guide or Acting Director are directed into the page.
4. **Exit condition:** Password is changed successfully or an error occurred.
5. **Flow of events:**
    - **5.1.** System shows the error message “New password can not be the same as the old one”.
    - **5.2.** System shows the standards of a password and asks the profile owner to retry.
    - **5.3.** If password requirements meet the standards and are different from the existing one, the new password is saved to the database and a success message is shown.
    - **5.4.** If password requirements do not meet the standards or same with the existing one, an error message is shown.
6. **Special /quality requirements:** This process can only be completed 5 times at a time for security measures.

1. **Name:** View Others’ Profile

2. **Participating actor:** Advisor, Acting Director, Coordinator
3. **Entry condition:** The dropdown menu for profiles is pressed.
4. **Exit condition:** One of the profile options is chosen.
5. **Flow of events:**
    - **5.1.** The profile option menu is pressed.
    - **5.2.** One of the profile options is chosen.

1. **Name:** View Guide’s Profile
2. **Participating actor:** Advisor, Acting Director, Coordinator
3. **Entry condition:** Guide’s Profile option from the dropdown menu is chosen.
4. **Exit Condition:** Viewing the profile.
5. **Flow of events:**
    - **5.1.** The Guide’s Profile option is selected.
    - **5.2.** All guides are listed and the system asks to select one of them.
    - **5.3.** Selected guide’s profile information is shown and schedules are shown.

1. **Name:** View Advisor’s Profile
2. **Participating actor:** Advisor, Acting Director, Coordinator
3. **Entry condition:** Advisor’s Profile option from the dropdown menu is chosen.
4. **Exit Condition:** Viewing the profile.
5. **Flow of events:**
    - **5.1.** The Advisor’s Profile option is selected.
    - **5.2.** All advisors are listed and the system asks to select one of them.
    - **5.3.** Selected advisor’s profile information and schedule are shown.

1. **Name:** View Counselors List
2. **Participating actor:** Acting Director and Coordinator
3. **Entry condition:** Clicking on “View Counselors List” button.
4. **Exit Condition:** Viewing the list or selecting a counselor.
5. **Flow of events:**
    - **5.1.** List of all counselors and highschool they represent are shown.

1. **Name:** View Dashboard for Data Collected
2. **Participating actor:** Acting Director and Coordinator
3. **Entry condition:** Clicking the “View Dashboard for Data Collected” button.
4. **Exit Condition:** Viewing the dashboard.
5. **Flow of events:**
    - **5.1.** Highschool data is listed which include their past visits to Bilkent, university entrance exam success etc.
    - **5.2.** Highschools are categorized into 4 groups.

1. **Name:** See Own Tour Schedule

2. **Participating actor:** Guide and Advisor
3. **Entry condition:** Clicking to “See Own Tour Schedule” button.
4. **Exit Condition:** View the schedule.
5. **Flow of events:**
   - **5.1.** When clicked to the button, the schedule of the profile owner is shown on the screen.
   - **5.2.** In the schedule, time slots occupied by tours are named with “In tour”.
   - **5.3.** If clicked to a “In tour”, information about the tour is shown.

3) Tech-Stack

Back-end: We are planning to use Java Spring Boot for our back-end. We chose this framework because it is scalable and secure. Also all of our team members are familiar with Java.
Front-end: We are planning to use the React library and JavaScript. We chose this library because it incorporates nicely with Java Spring Boot, and it is easy to use.
Database: We are planning to use MongoDB for our database. We chose this database because it works smoothly with Spring using Spring Data MongoDB.