

CS 319-1 Group 8

Deliverable 1 - 1st Iteration

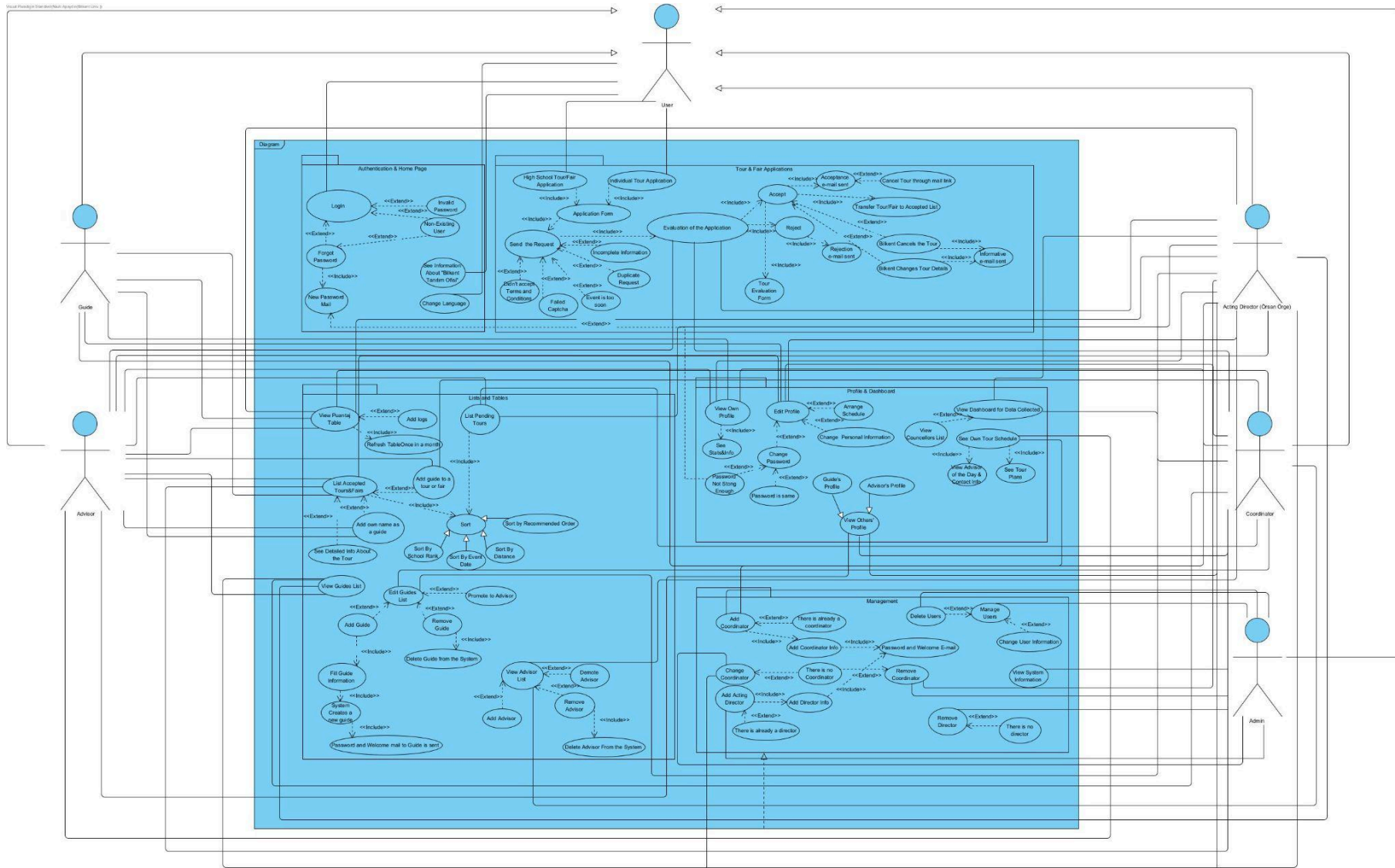
Bilkent Tanıtım Ofisi Management System

Short-name: BILFO

11.10.2024

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 could include 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 a few 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.
 6. **Special /quality requirements:** This process can only be completed a few times at a time for security measures.
-
1. **Name:** New Password Mail
 2. **Participating actor:** User
 3. **Entry condition:** User successfully uses "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.

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:** The system does the necessary checks and if everything is okay a way to enter the information of the new coordinator opens.
5. **Flow of events:**
 - 5.1. Admin or Director clicks on the add coordinator button.

1. **Name:** Add Coordinator Info
2. **Participating actor:** Admin or Director
3. **Entry condition:** Admin or Director clicked the add coordinator button and there is no issue with adding a coordinator.
4. **Exit condition:** A new coordinator is added into the system.
5. **Flow of events:**
 - 5.1. Admin or Director clicks on the add coordinator button.
 - 5.2. Admin or Director enters the required information for the coordinator.
 - 5.3. A new coordinator is added into the system with the provided information and the system moves to "Password and Welcome Email".

1. **Name:** Password and Welcome Email
2. **Participating actor:** Admin or Director
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.

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:** The system does the necessary checks and if everything is okay a way to enter the information of the new director opens.
 5. **Flow of events:**
 - 5.1. Admin clicks to the Add Coordinator button.
 - 5.2. The system checks if a Director exists, if it does, this continues to "There is already a Director".
 - 5.3. The director is added to the system.
-
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:** Add Director Info
 2. **Participating actor:** Admin

3. **Entry condition:** Admin successfully enters the process to add a director.
4. **Exit condition:** The director with the entered information is created.
5. **Flow of events:**
 - 5.1. Admin enters the necessary information for the new director.
 - 5.2. A new director is created, and the system moves to the "Password and Welcome E-mail".

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:** Manage Users
2. **Participating actor:** Admin
3. **Entry condition:** Admin clicks the button for managing users.
4. **Exit condition:** A list of all users and ways to manage them appears for the admin.
5. **Flow of events:**
 - 5.1. Admin clicks the button to manage users.
 - 5.2. A list of all the users and the ways to manage them appears. This includes deleting them and changing their information.

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:** 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.
-
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.
-
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. From the most recent to the followings, the list is shown to the user.
-
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 most far away 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 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 the our algorithm we will implement.
 - 5.3. From the most urgent to the lowest schools, the list is shown to the user.

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.

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.

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:** Edit Guides List
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** Acting director or coordinator clicks to the Edit Guides List button.
4. **Exit Condition:** All guides in the system are shown through a list.
5. **Flow of events:**
 - 5.1. Acting Director or coordinator click to the Edit Guides List button
 - 5.2. Editable Guides List is shown.

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:** Acting director or coordinator is directed to the fill guide information page.
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.

1. **Name:** Fill Guide Information
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The Fill Guide Information page is shown
4. **Exit Condition:** The data is sent to the database
5. **Flow of events:**
 - 5.1. Fill Guide Information page is shown.
 - 5.2. Acting Director or Coordinator enters the required information.
 - 5.3. The data is sent to the database.

1. **Name:** System Creates a New Guide
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The data from the Fill Guide Information page is sent to the database.
4. **Exit Condition:** The data is stored in the database
5. **Flow of events:**
 - 5.1. The data is sent to the database
 - 5.2. The data is stored in the database.

1. **Name:** Password and Welcome Mail to Guide is Sent
2. **Participating actor:** Director, Coordinator
3. **Entry Condition:** The data is stored in the database.
4. **Exit Condition:** Password and welcome mail is sent to the new guide.
5. **Flow of events:**
 - 5.1. The data is stored in the database.
 - 5.2. The password and welcome mail is sent.

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

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

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 "Acceptance Email Sent".

5.6. If rejected, a rejection email will be sent.

1. **Name:** Acceptance Email Sent

2. **Participating actor:** Advisor, Acting Director, Coordinator and User

3. **Entry condition:** Advisor, Coordinator or Acting Director accepts a request.

4. **Exit condition:** E-mail is sent.

5. **Flow of events:**

5.1. Advisor, Coordinator or Acting Director accepts a request.

5.2. An email that says "the request is accepted" will be sent to the user. The mail will also include a link that allows users to cancel the tour.

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 cancelled 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:** Tour Evaluation Form
2. **Participating actor:** User
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:** Send the Request
2. **Participating actor:** User
3. **Entry condition:** User enters the website and opens the application form button.
4. **Exit condition:** User completes the application, either successfully or unsuccessfully.
5. **Flow of events:**
 - 5.1. The user opens the website and opens the application form.
 - 5.2. The user selects the application type, this continues on the "Application Form"
 - 5.3. The user fills out the necessary information.
 - 5.4. The user reads and accepts the Terms and Conditions or else the application will fail.
 - 5.5. The user 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. 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.

1. **Name:** Application Form
2. **Participating actor:** User
3. **Entry condition:** User opens the application form.
4. **Exit condition:** User saves his/her choice.
5. **Flow of events:**
 - 5.1. The user selects the application type, which is either a High School Tour/Fair Application or an Individual Tour Application.
 - 5.2. The necessary information will change according to the application type.

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:** This process can only be completed a few times at a time for security measures.

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 a few 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:** 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:** 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 Councillors List
- 2. **Participating actor:** Acting Director and Coordinator
- 3. **Entry condition:** Clicking to "View Councillors List" button.
- 4. **Exit Condition:** Viewing the list or selecting a councillor.
- 5. **Flow of events:**
 - 5.1. List of all councillors 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.

- 1. **Name:** View Advisor of the Day & Contact Info
- 2. **Participating actor:** Guide and Advisor
- 3. **Entry condition:** Clicking the "View Advisor of the Day & Contact Info" button.
- 4. **Exit Condition:** Viewing the advisor of the day and their contact information.
- 5. **Flow of events:**
 - 5.1. When clicked on the button, the advisor of the day and their contact information are shown on the screen.

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.