Bilkent University

Department of Computer Engineering

CS 319 Term Project

Fall 2024

Section 1

Group 4

Deliverable 2

1st Iteration,

Group Members:

Berin Su İyici - 22102342

Arda Kırcı - 22002031

Ali Sher Mir - 22201310

Alper Yıldırım - 22102033

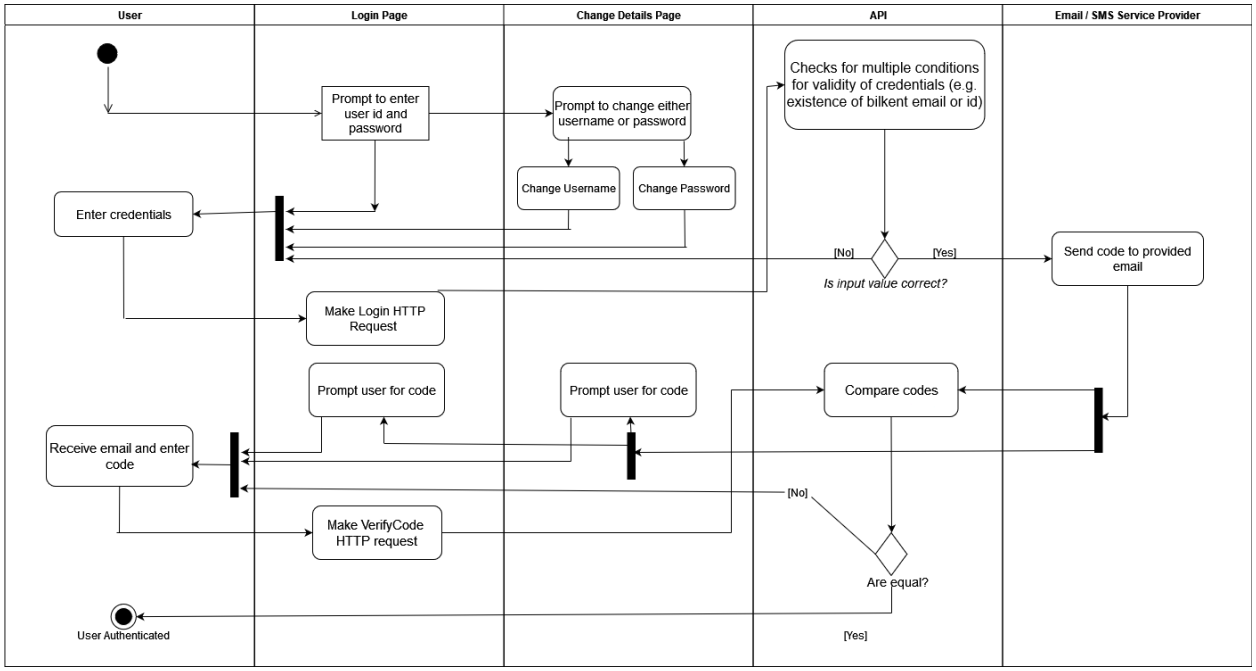
Melih Rıza Yıldız - 21902958

Instructor: Eray Tüzün

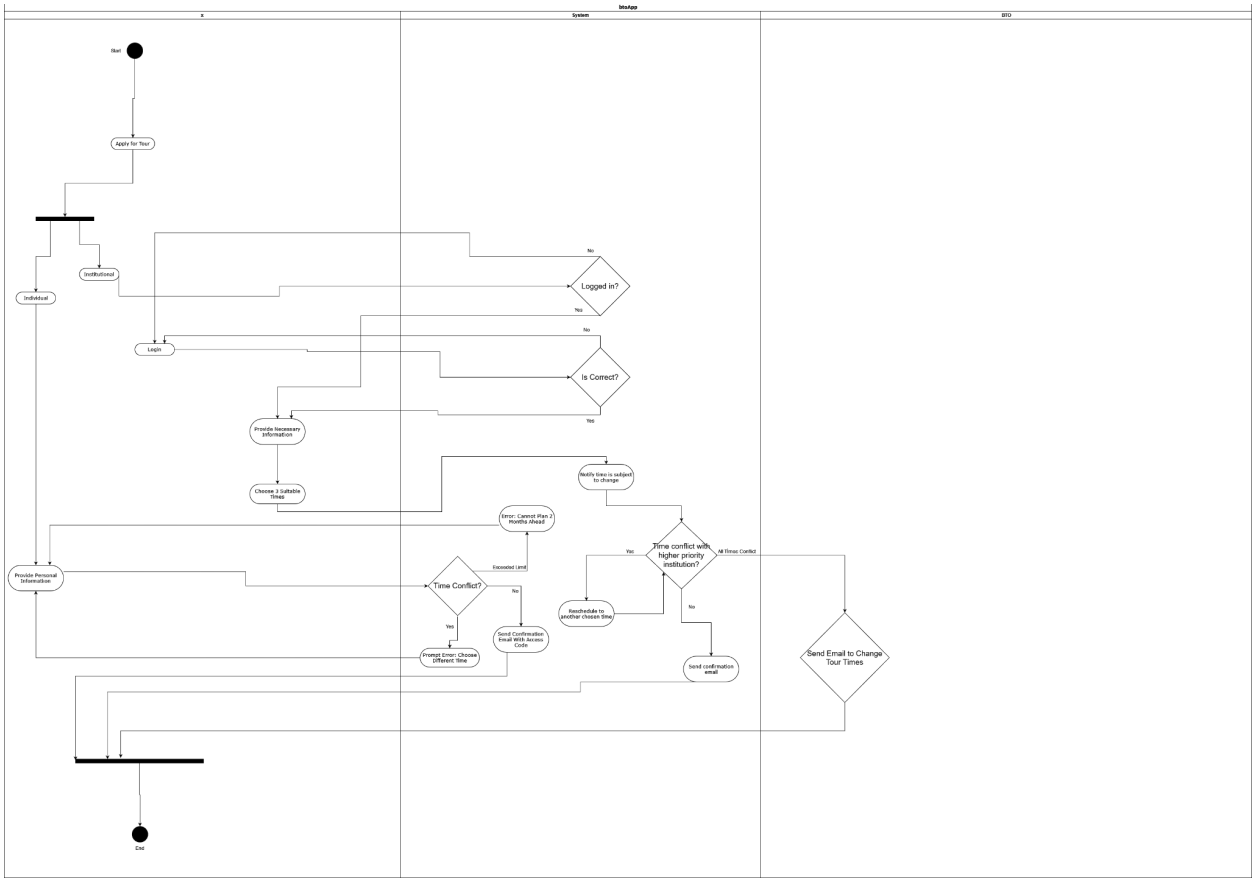
Teaching Assistant: Yahya Elnouby

Following areas are covered in the use case diagram:
- **Authentication System:** Includes login, account management, and authentication.
- **Tour System:** Covers schools, tours, forms, and appointments.
- **Guide Management:** Involves adding, removing, and managing guides.
- **Feedback System:** Includes managing user feedback.
- **Appointment Management:** Details for scheduling, changing, and managing appointments.
- **Analytics System:** Includes timesheets of the guides and monthly reports.

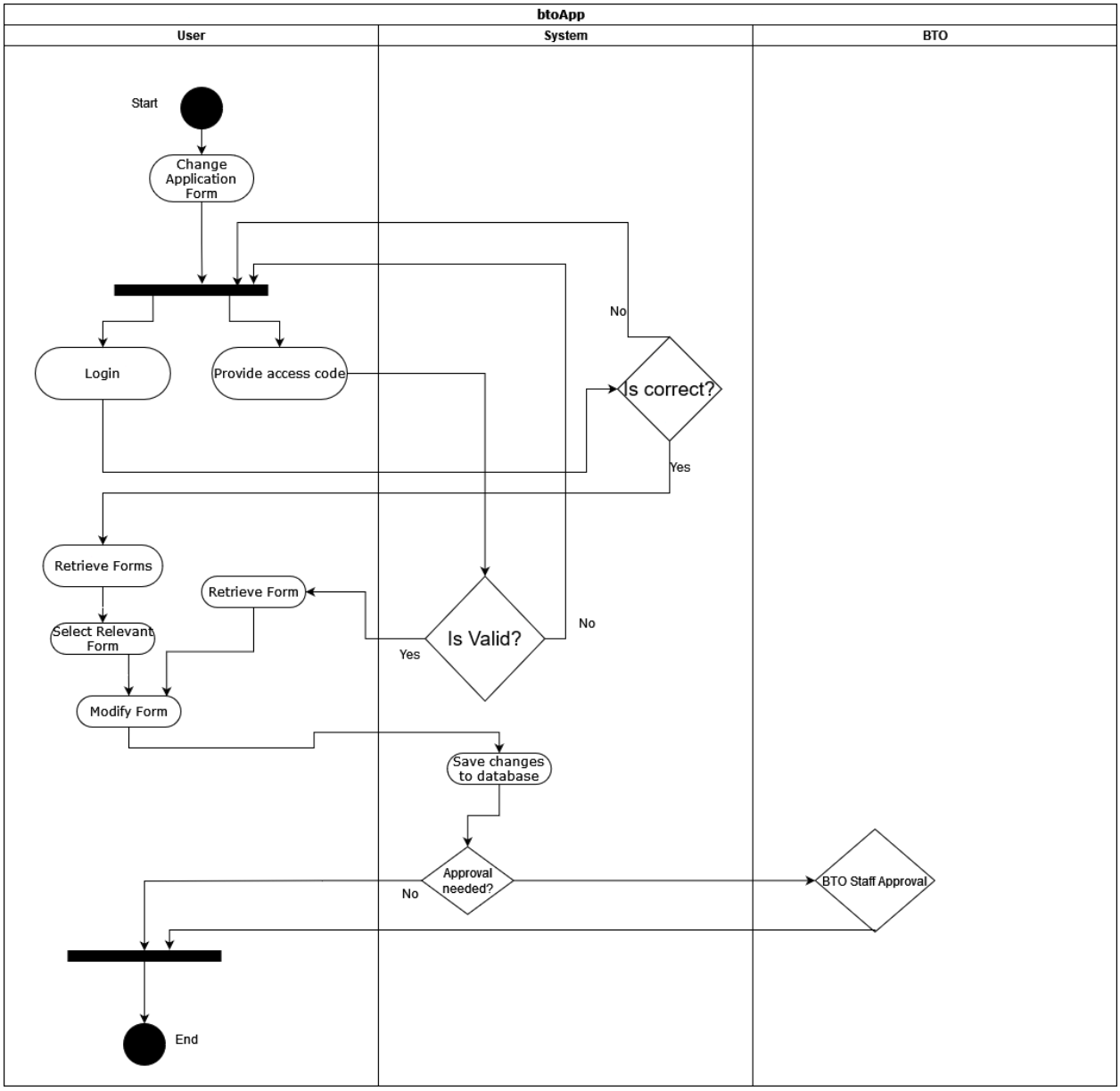
1. Activity Diagrams
    - User Account & Authentication Activity Diagram:**
      Combines login, account management, and authentication => user account workflows.



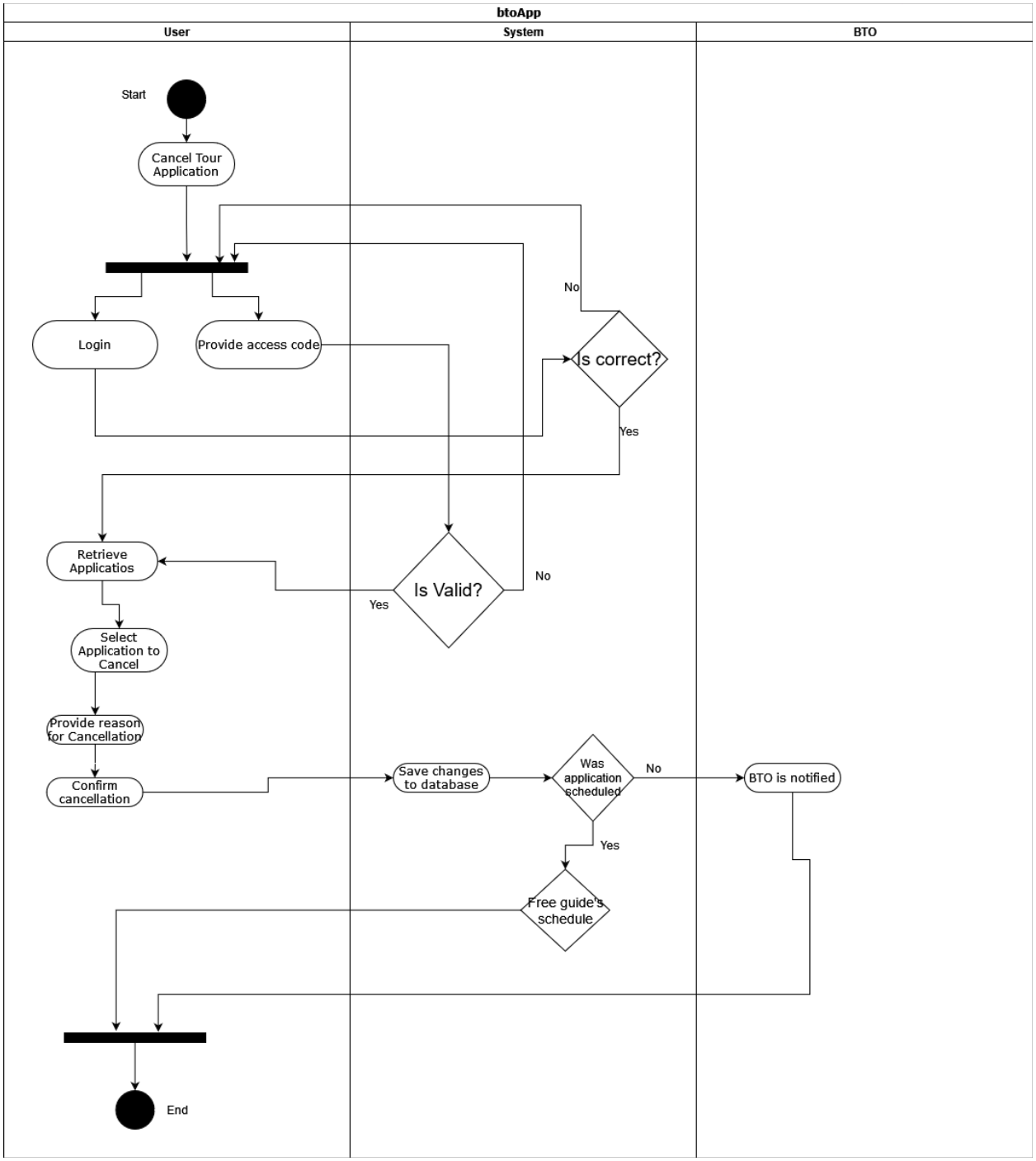
- Tour Application & Appointment Management Activity Diagram:**
  Diagram covers creating, changing, and managing appointments. A combined activity diagram for both individual and institutional applications, including booking, scheduling, and changing appointments.



tourManagement Activity



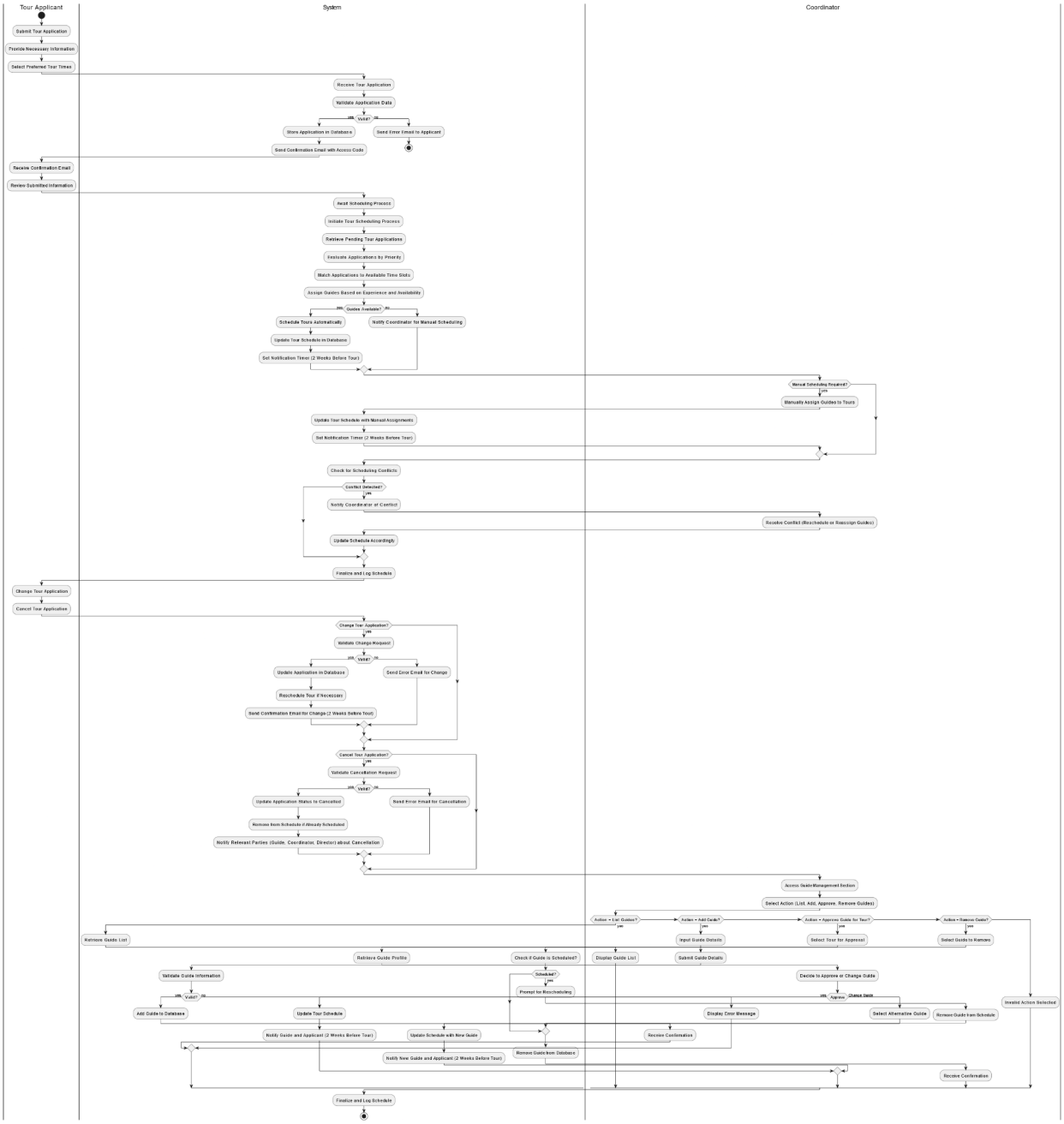
changeApplication Activity



cancelTourApplication Activity

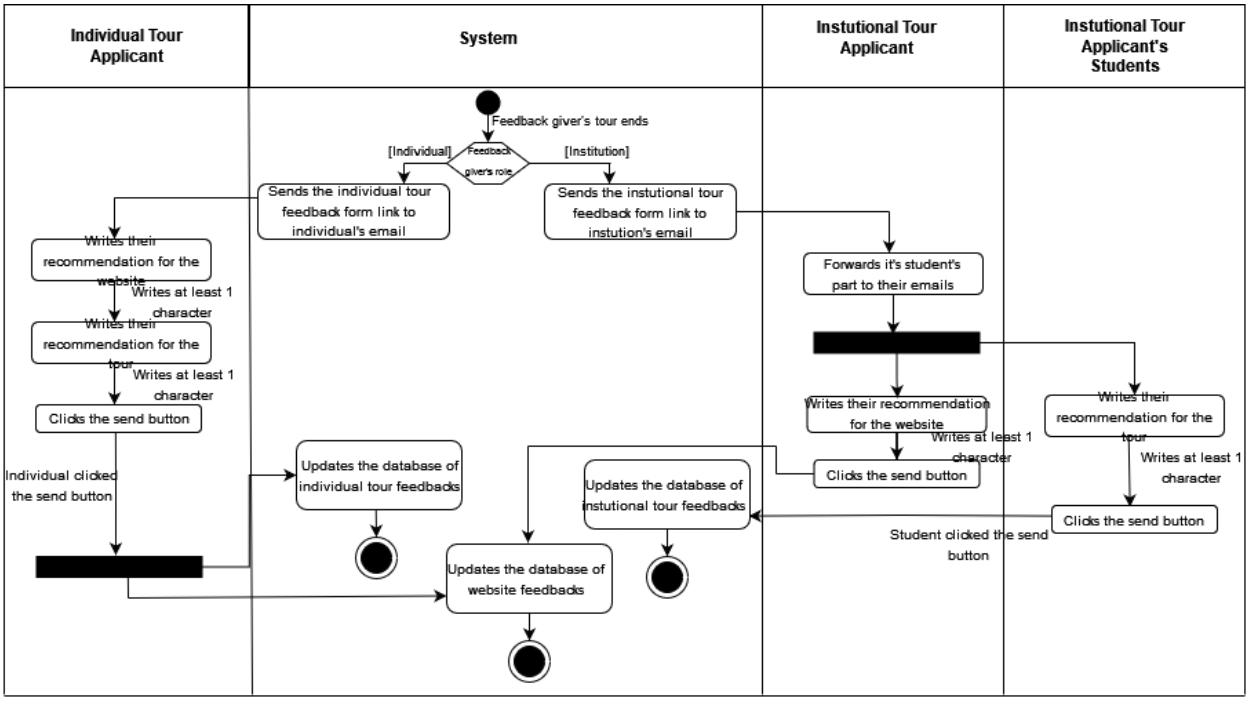
- Guide Management & Tour Scheduling Activity Diagram:**

Adding, removing, and scheduling guides are shown in the same diagram, a single activity diagram can represent guide management along with scheduling/rescheduling tours.



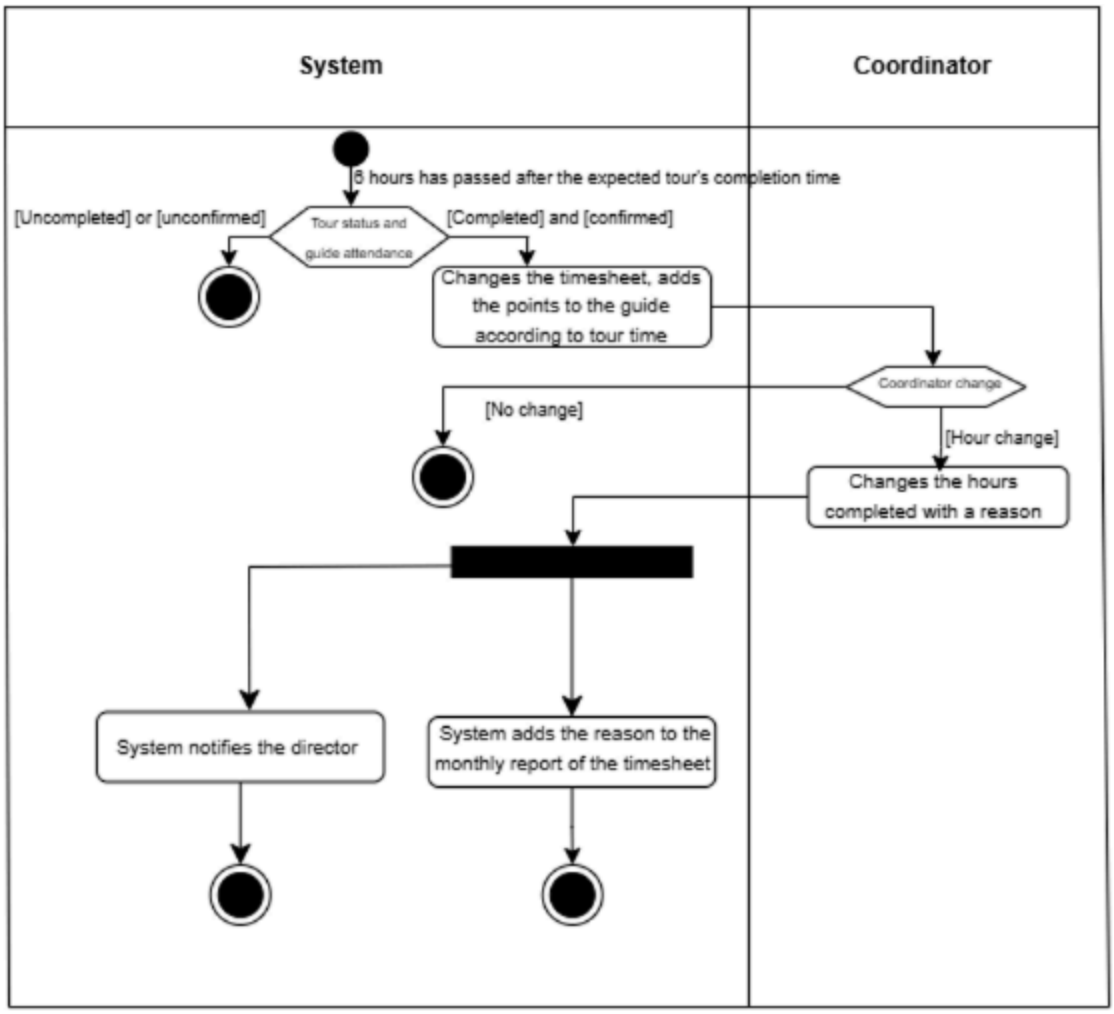
- Feedback Activity Diagram:**

Diagram covers feedback submission and processing. A combined activity diagram for both individual and institutional tour's feedback and also about the webpage, which will be sent to the system.



- Analytics Activity Diagram:**

Diagram covers analytics for the timesheets of the guides and monthly reports. A combined activity diagram for both unaccepted and accepted tours. Coordinators can change the hours guides did with a reason to the system, which will be included in the monthly report.



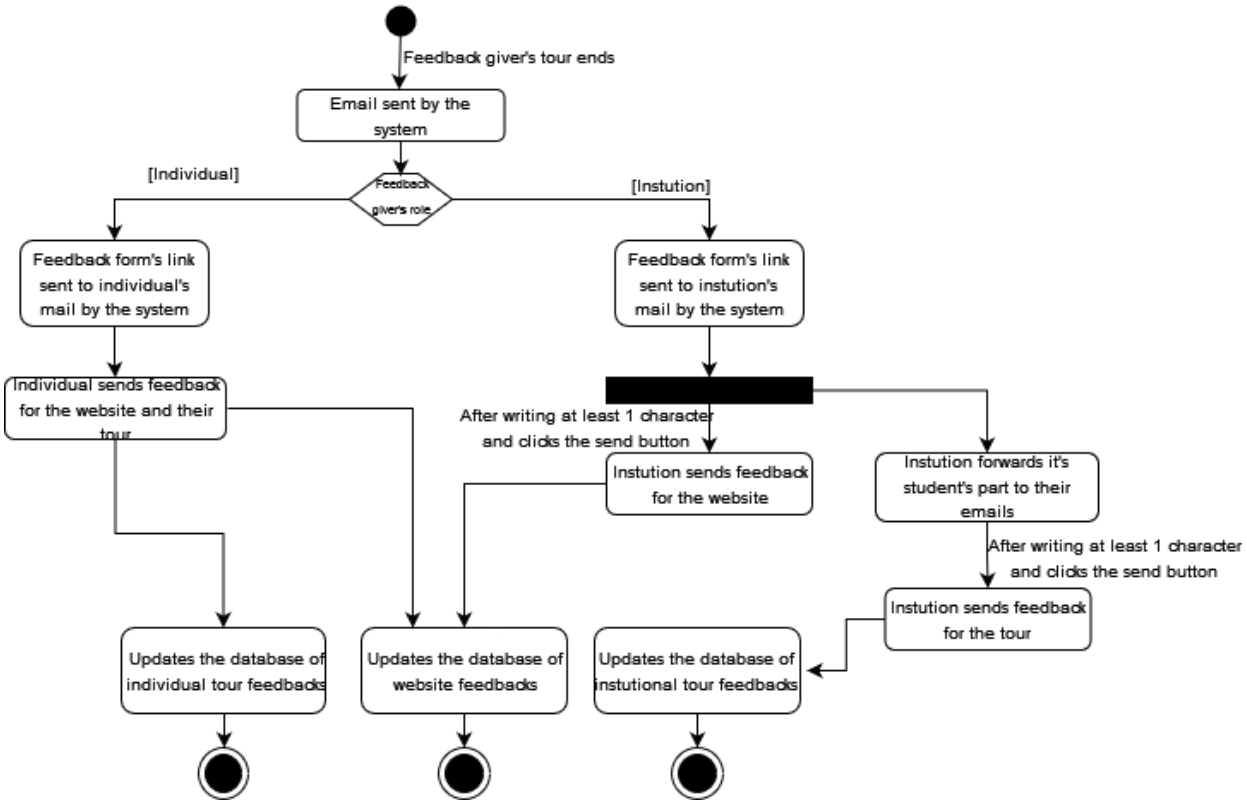
2. State Diagrams

- User Account & Authentication State Diagram:

- Since the use case groups authentication processes with user account management, a single state diagram can represent the states of user accounts, covering login, logout, and account updates.

- Tour Application State Diagram:

- This diagram focuses on the states of the tour applications: new, pending, approved, scheduled, completed, and canceled.



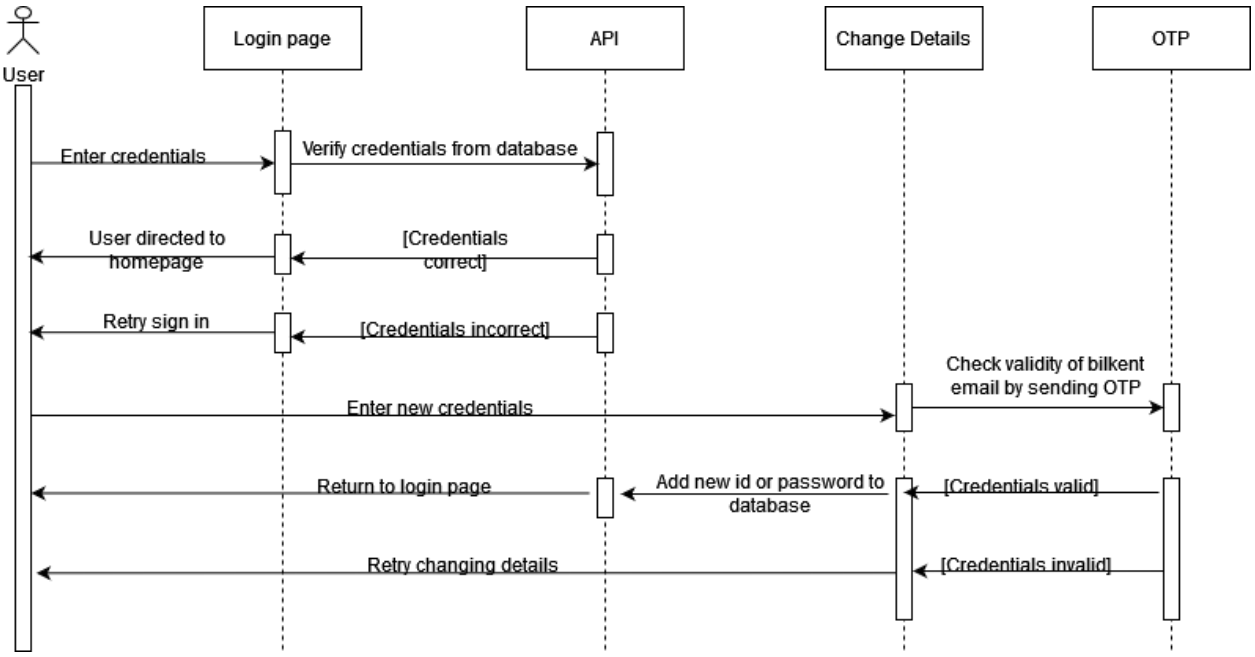
- Guide State Diagram:

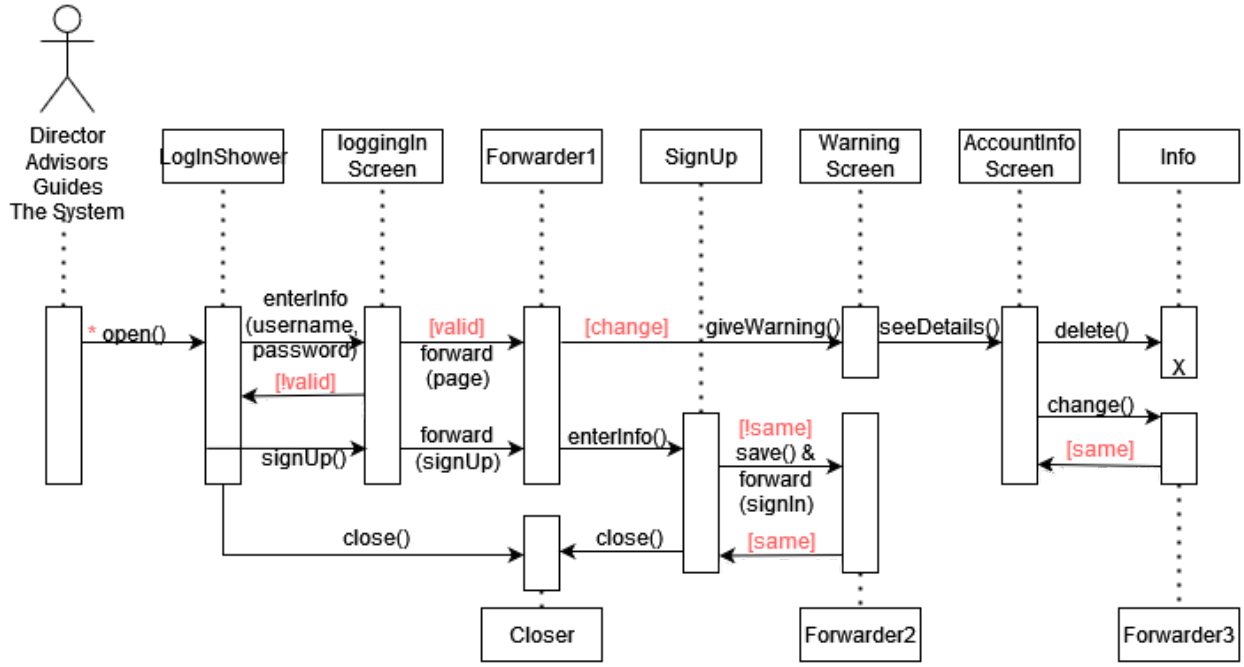
- A separate state diagram for guide management is still useful, showing states like trainee, active guide, and unavailable guide.

3. Sequence Diagrams

- User Interaction Sequence Diagram:

- This can cover the sequence for login, account creation, and account management in one diagram


User
Login page
API
Change Details
OTP
Enter credentials
Verify credentials from database
User directed to homepage
[Credentials correct]
Retry sign in
[Credentials incorrect]
Check validity of bilkent email by sending OTP
Enter new credentials
Return to login page
Add new id or password to database
[Credentials valid]
Retry changing details
[Credentials invalid]


Director
Advisors
Guides
The System
LogInShower
loggingIn Screen
Forwarder1
SignUp
Warning Screen
AccountInfo Screen
Info
* open()
enterInfo (username, password)
[!valid]
[valid] forward (page)
signUp()
forward (signUp)
[change]
enterInfo()
giveWarning()
[!same] save() & forward (signIn)
[same]
seeDetails()
delete()
X
change()
[same]
close()
close()
Closer
Forwarder2
Forwarder3

Non-Functional Requirements

Maintainability

- Project's codes will follow consistent coding standards. To name a few, naming conventions, clear commenting, and consistent structures. For naming conventions, camel case will be used for variables, APIs will be written in screaming snake case.

- Project will be divided into different parts according to functionalities, such as feedback, analytics. This will also make the updates easier to implement, otherwise it would make the code unnecessarily complex.

- Code will be tested again and again while writing and also after writing to cover all the bugs, and decrease the unexpected error amount to at most 5. Expected errors will be handled. Even if that error was not handled, the page will be directed according to the site's error code.

By following these rules, this project will be effortlessly updated, and modified even by the teammates,

Usability

- At least 40% of the code lines will be explained by comments. For consistent structures, all indentations will be 4 spaces, all UI components will be in the "components" folder but the page's layout will be in the "pages" folder. This will make the code readable not only for the writer but also to the other members of this project.
  If an applying institution, whose preferred tour times have already been filled by other schools, does not make a change, their tour application will be deleted 2 days before the time they chose for their latest chosen tour time.

Reliability

- **Safety**: There will be random alphanumeric code to edit or delete information, which is special for any tour.
  Some parts (guide list, manage account) will be reached by only the admin.

Performance

- **Response Time**: The application is expected to be highly responsive, with a target response time of under 5 seconds for most user interactions, ensuring a smooth user experience.
- **Scalability**: The system must possess robust scalability features to effortlessly accommodate the increasing number of users and items within the marketplace as it grows.
- **Load Handling**: It is imperative for the application to efficiently manage heavy loads, capable of seamlessly handling a large influx of concurrent users without experiencing performance degradation.

Mockup

https://app.uizard.io/p/413a77ac