Bilkent University

Department of Computer Engineering

CS319 Term Project

Section 1

Group 1A - "lobby"

Project short-name: Pandemica

Project Analysis Report

Group Members:

- Efe Beydoğan (21901548)
- Arda Önal (21903350)
- Mert Barkın Er (21901645)
- Emir Melih Erdem (21903100)
- Gökberk Beydemir (21902638)
- Eren Polat (21902615)

Table of Contents

Introduction

We have decided to make our pandemic manager a software product that will ease everyone's lives in the university during difficult pandemic times. To this end, we will combine every resource a person might need in one place, and provide easy access to all of the information that could be detrimental in handling a pandemic. The project will be a medium for students, lecturers, TAs and health personnel to get together during rough times to inform each other of the latest developments regarding the pandemic. This will allow the members of the university to take precautions together and ensure a healthy and happy experience for everyone.

The product will feature the personal information of the user as well as weekly reports, guidelines specified by the university, important announcements, statuses of other people in the user's registered courses, and much more. Also, by adding a feedback form we are hoping to receive feedback from the users about the app and increase service quality. The project will include a request form, which users can fill out and send to the university to make suggestions on pandemic related precautions (addition of disinfectants to certain places etc.). Our goal when developing this product is to save the inhabitants of the university from the hassle of finding every bit of information separately on the internet, and let everyone have easy access to crucial information. As a result, the app will hopefully help alleviate the harsh circumstances of a pandemic and let people focus on their studies without much worry. The project will have the following basic features:

- Display information about the user (Name, ID, COVID status, test results if available, vaccination card if the user has uploaded one)
- Sending notifications based on changes in the COVID status of either the user or their friends/classmates
- University wide stats (1st dose/2nd dose percentage, total number of doses, number of applied doses (1st/2nd dose), number of tests, number of current cases)
- Weekly COVID report
- Showing important announcements and guidelines set by the university
- Violation report system so users can report violations they observe in the campus
- Feature to add friends so the user can track their friends' statuses as well
- Getting Diagnovir test appointment

Proposed System

Overview

This software is a general pandemic manager for Bilkent University. After logging in, a user will be greeted with the main menu containing access to personal information, general information, report & request and social pages.

Personal Information

All users can access their personal information. Users can upload and view their HES codes and vaccination cards here. After they upload their HES codes and vaccination cards they will be able to view their vaccination information, such as which types of vaccinations they had administered, how many shots they had and when they had those shots, they can also see their QR code. Users can see their own personal information as well, which includes their SRS picture, their Bilkent ID number etc. Students can select their neighbors in a section. Admins can send notifications to users if they decide to make an important announcement. Users can view their test results, and can see if their test result is positive or not. Admins can view and modify user information. Users can view their own sickness or risk status, they can view a notification and delete it after viewing that notification. Admins can view every neighbor of a user or only view the healthy/risky neighbors of a user. Health Center Employees can update a user's risk status and upload physical examination results, they can also enter the results of those tests. Students can select their seats for a section, which the instructor will be able to see when they choose to view a section's information. Users can make test appointments.

General Information

Users can view the university wide stats, which shows the dose percentages in the university, total number of tests conducted, number of current cases etc. They can view guidelines Bilkent University put in place to prevent the further spread of this pandemic. They can view the weekly report that contains the number of current cases divided by their type (academic staff, admin staff, students and employees), the number of new cases and the number of people recovered in the last week. They can choose to view the previous weeks' or this week's reports in this page as well. They can view the announcements made by the Bilkent University about the pandemic here, like changing a building's opening hours to help prevent more spread around that area.

Admins can create and update the university wide stats. They can create and update the guidelines made by the university. They can create and update the announcements made by the university. They can create and update the weekly reports.

Report and Request

Users can create violation reports to report someone they saw that was violating the rules of Bilkent University on pandemic prevention. They can view their previously created reports that were not deleted before as well. Users can create request forms to request from the university a pandemic related need, such as adding a hand sanitizer to the entrances of the buildings or refilling the said sanitizers. Users can create feedback forms, in which they can either request a new feature to be added to the project or just state their general feedback about the site. They will be able to view their previously created forms as well.

Admins can view all created request forms, feedback forms and violation reports. After viewing a report or a form, they can choose to make a decision about that form or report. After that they can delete that form or request.

Social

Students can send friend requests to any other student, can accept those requests and remove a friend from their friend list. Students can view all of their friends, see their information and filter their sickness and/or risk status.

Functional Requirements

- **Logging in and signing up**

Users and Admins must be able to login and sign up with their ID’s and passwords. They should only be able to view certain pages and options depending on their role.

- **HES Code and Vaccination Cards**

The users must be able to upload their HES Codes and their vaccination cards to the website without problems. If they upload an invalid file, the file they uploaded should not be saved to their account and they should be prompted to try again. The admins must be able to examine the uploaded HES Codes at any time to check if a user is at risk, is currently sick or if the user is allowed to come to the campus. The admin must also be able to look up users’ vaccination cards and extract the information from there and make decisions based on that. One example for this is that if a user gets in touch with someone that is sick, if they have two shots of vaccination and their test result comes back negative, they should be allowed into the campus.

- **Seating Plans**

The academic personnel must be able to create a seating plan for a classroom and update it accordingly. Students must be able to choose their seats in their sections and view their seating plan. If a student gets sick or gets risky, their neighbors should be sent a notification warning them and their health status should be set to risky. Admins must be able to view these classroom seating plans and send notifications to the students who sit close to the students who get sick.

- **Personal Information**

Users must be able to view their own personal information and their test results in a page. Admins must be able to view any user’s information and edit that information if necessary.

- **General Information**

Users must be able to view general stats about the university, the guidelines to contain the spread of the virus and the announcements made by the university. Admins must be able to create and update those things.

- **Reports and Requests**

Users must be able to create and file violation reports, request forms and feedback forms. They should see their previously created forms and reports as well. Admins must be able to see the reports and requests made by all users, view a specific request/report and delete it after handling the report.

- **Social**

Users should be able to add friends by sending friend requests and remove friends if they want to. They should be able to view which of their friends are at risk and which of them are safe.

- **Notifying Instructors**

In case one of the students in one of their sections gets sick, the instructors should be immediately notified by the system. The notification should include the particular student’s information for the teacher to consult and make sure they are not in the classroom. The instructor should also be notified right before their class with the specific section starts, so they don’t have to dig through their emails to find the name of the students that they are not supposed to let in.

- **Making test appointment**

Every user in the system should be able to make a test appointment to be carried out at Bilkent University.

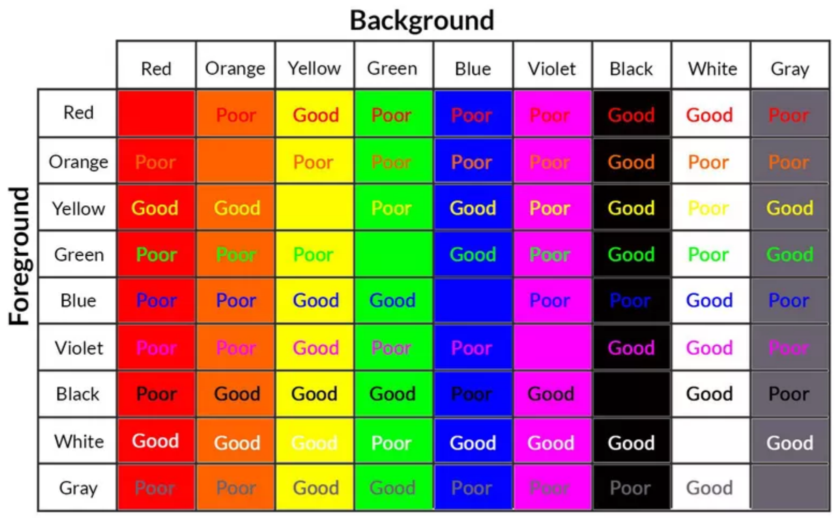
Non-Functional Requirements

Performance

- The system website should load in less than 2 seconds.
- Every user interaction should be processed and if there is output, the result should be displayed in less than 2 seconds.
- The time that takes for the website to retrieve information from the database should be less than 1 second.
- The HES code should be revalidated every 30 seconds.
- If there is a change in the covid status of a user, the change should be represented in less than 2 seconds.
- The course and section information should be validated every 30 seconds and if there is any change, it should be displayed in less than 2 seconds.
- The entire system data should be backed up every hour.

Usability

- Any user who knows how to read and write should be able to use the system without a user manual.
- The buttons should be aligned.
- Every button that is visible to the user should be clickable.
- The colors of the background and the foreground labels that are in the user interface should according to Fig. 1.



Foreground	Background								
	Red	Orange	Yellow	Green	Blue	Violet	Black	White	Gray
Red		Poor	Good	Poor	Poor	Poor	Good	Good	Poor
Orange	Poor		Poor	Poor	Poor	Poor	Good	Poor	Poor
Yellow	Good	Good		Poor	Good	Poor	Good	Poor	Good
Green	Poor	Poor	Poor		Good	Poor	Good	Poor	Good
Blue	Poor	Poor	Good	Good		Poor	Poor	Good	Poor
Violet	Poor	Poor	Good	Poor	Poor		Good	Good	Poor
Black	Poor	Good	Good	Good	Poor	Good		Good	Poor
White	Good	Good	Good	Poor	Good	Good	Good		Good
Gray	Poor	Poor	Good	Good	Poor	Poor	Poor	Good	

Fig. 1: Table representing how the background and foreground colors should be picked.

Reliability

- The information displayed should always be correct.
- A system crash should not result in any data loss.
- Reports that the user generates should not be lost due to the user's internet connection loss.
- The system will have confidential information of every user such as passwords, phone numbers, Bilkent ids, email etc. and this information should not be accessible to anyone who tries to obtain them.
- Any functionality of the website should work regardless of cybersecurity attacks.
- The system should not crash in input errors and inform the user that the input is incorrect.

Supportability

- The website should be supported for Google Chrome version 68.x or higher, Opera version 80.x, Safari 10.x higher or higher and Microsoft Edge version 93.x or higher.
- The locations of labels and buttons should not change for every operating system and browser.
- The website should work on devices that have 2 GB RAM or higher.
- The database should be able to store data that is up to 1 T.

System Models

Use Case Models

Note: The word “Actor” in all of the use case explanations refers to all participating actors in a particular use case.
Note 2: The word “User” in all of the diagrams and use case explanations refers to the given hierarchy below:

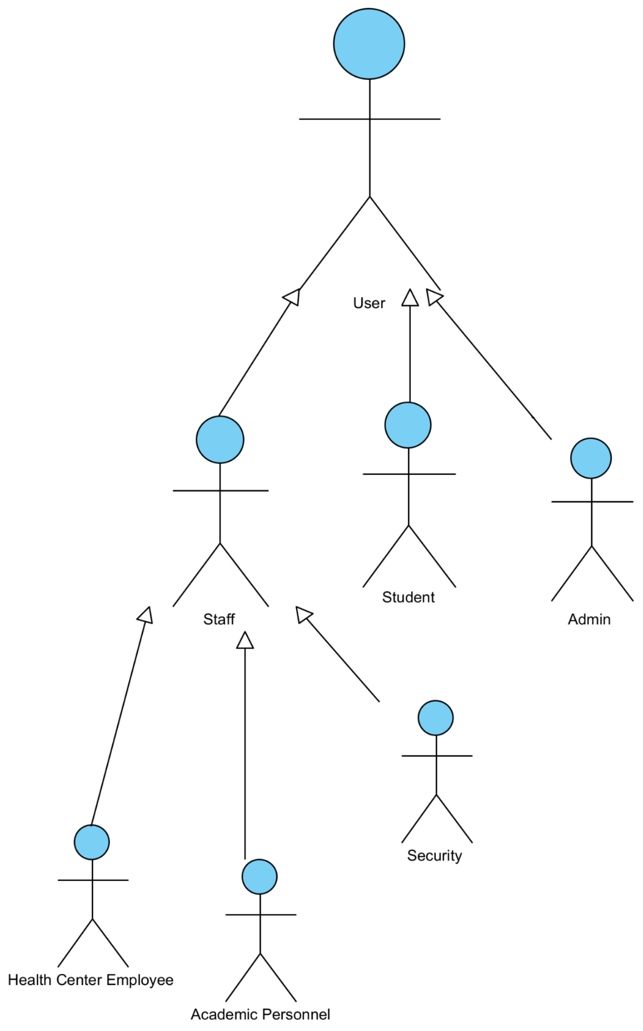


Fig. 2: Actor hierarchy

Use Case Diagram

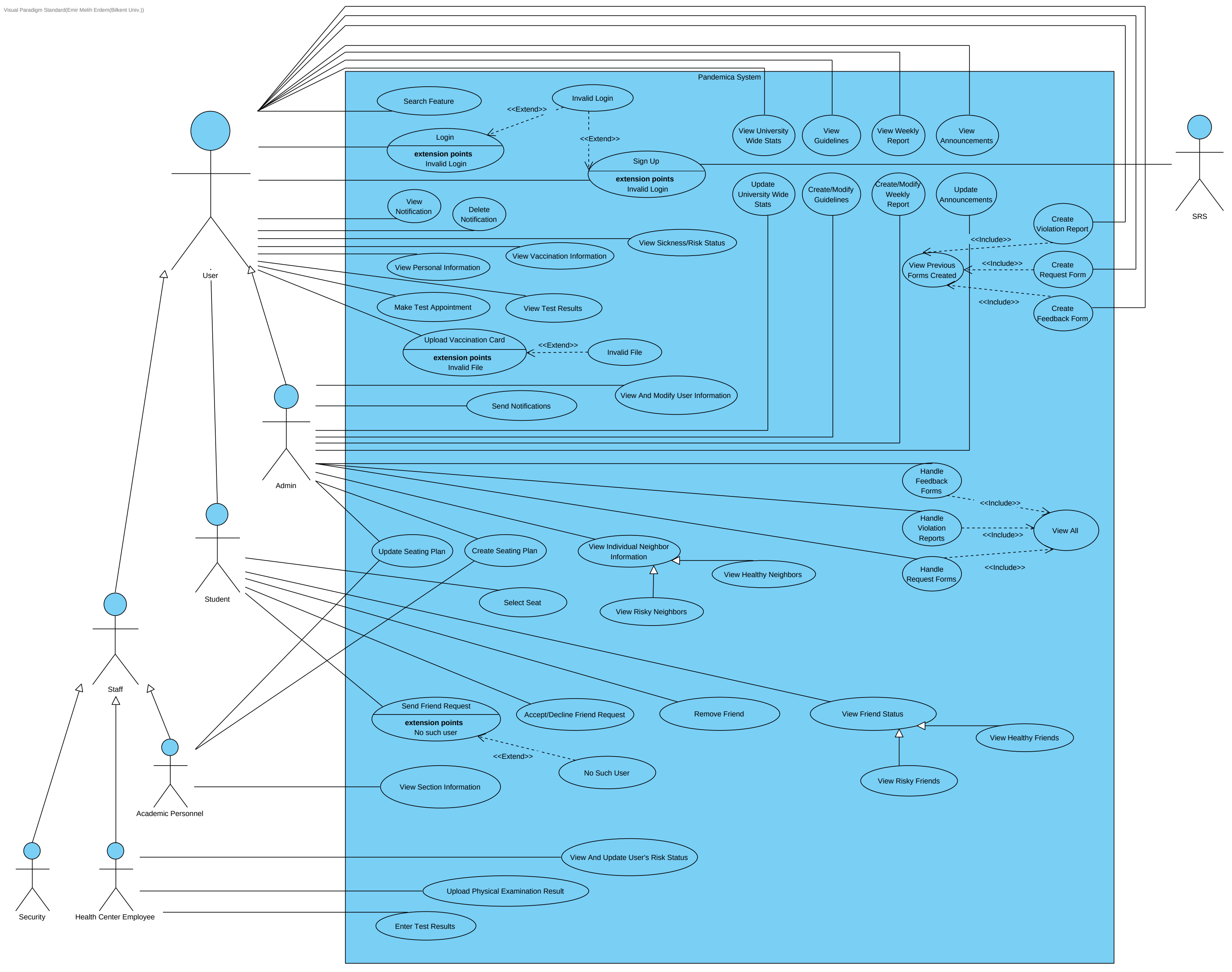


Fig. 3: Use Case Diagram for Pandemica System

Use Case: Login

Participating Actor(s): User

Entry Conditions: Actor isn't logged in and already signed up.

Exit Conditions: Actor enters correct id and password

The Flow of Events:

- Actor clicks on id/password fields
- Actor enters information
- If information is valid actor logs into the system

Use Case: Invalid Login

Participating Actor(s): User

Entry Conditions: This use case extends the Sign Up and Login use cases. It is initiated by the system whenever the user enters invalid information in one or more of the input fields.

Exit Conditions: User enters valid information

The Flow of Events:

- Actor tries to login or sign up
- Actor enters invalid information in one of the input fields
- The system displays an error message and the actor has to re-enter the faulty information

Use Case: Sign Up

Participating Actor(s): User, SRS

Entry Conditions: User clicks on sign up button

Exit Conditions: User submits the information for signing up

The Flow of Events:

- User decides to sign up to the site
- User fills out the sign up form with correct information
- SRS validates the information supplied by the user

- User submits the information and signs up

Use Case: Search Feature

Participating Actor(s): User, Admin

Entry Conditions:

- Actor is logged in
- Actor navigates to search bar

Exit Conditions: Actor either selects a search result or cancels the search

The Flow of Events:

- Actor clicks on the search bar
- Actor types a search query for searching in the website
- Actor gets a list of the results
- Actor clicks on one of the results or cancels the search

Use Case: Update Seating Plan

Participating Actor(s): Admin, Academic Personnel

Entry Conditions: Actor decides to enter the page for updating a seating plan

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor updates an existing seating plan

Use Case: Create Seating Plan

Participating Actor(s): Admin, Academic Personnel

Entry Conditions: Actor decides to enter the page for creating a seating plan

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor creates a new seating plan for a classroom

Use Case: Select Neighbor

Participating Actor(s): Student

Entry Conditions:

- Actor is logged in
- Actor navigates to the select neighbor screen to mark their neighbors in their classes

Exit Conditions: Actor leaves the select neighbor interface

The Flow of Events:

- Actor activates the functionality to mark their neighbors
- Actor is presented with a list of their classes and the students in each of the sections
- Actor selects the students sitting in their close proximity during classes
- Actor leaves the select neighbor screen

Use Case: View Vaccination Information

Participating Actor(s): User

Entry Conditions:

- Actor is logged in
- Actor navigates to vaccination information screen

Exit Conditions: Actor leaves the view vaccination information interface

The Flow of Events:

- Actor navigates to the vaccination information page
- Students and instructors view their individual vaccination information including the dates of the doses received, as well as the type of the vaccine. These roles also see their own official vaccination card, if they uploaded one on the system.
- Health Center Employees and admins view every registered user's vaccination information on the system via this functionality.
- Actor leaves the vaccination information screen

Use Case: View Personal Information

Participating Actor(s): User

Entry Conditions:

- Actor is logged in
- Actor navigates to the personal information screen

Exit Conditions: Actor leaves the view personal information screen

The Flow of Events:

- Actor navigates to the personal information page
- Students and instructors view their own personal information on this page. The personal information includes their Bilkent ID, TC ID/Passport Number, address, phone number, email etc.
- Health center employee and admin roles view every registered user's personal information on this page.
- Actor navigates out of the page

Use Case: Send Notifications

Participating Actor(s): Admin

Entry Conditions:

- Admin is logged in
- Admin navigates to the interface for sending notifications

Exit Conditions: Admin leaves the page

The Flow of Events:

- Admin decides to send an important notification regarding the pandemic and activates the send notification page
- Admin fills out a notification message
- Admin selects who they want to send the message to
- Admin sends out the notification
- Admin leaves the send notifications page

Use Case: View Test Results

Participating Actor(s): User

Entry Conditions:

- Actor is logged in
- Actor navigates to the page for viewing their test results

Exit Conditions: Actor navigates out of the page

The Flow of Events:

- Actor enters the interface for viewing test results.
- Students and instructors view their individual test results for either the PCR or the Diagnovir test.
- Health center employee and admin roles view every registered user's test results available on the system.
- Actor leaves the page

Use Case: View And Modify User Information

Participating Actor(s): Admin

Entry Conditions:

- Admin is logged in
- Admin enters a user's page to modify their information

Exit Conditions: Admin leaves the user's page after modifying their information

The Flow of Events:

- Admin opens a user page to view and modify the information that includes personal information and pandemic related information.
- Admin modifies the information as necessary or doesn't alter anything
- Admin saves the information and leaves

Use Case: View Sickness/Risk Status

Participating Actor(s): User

Entry Conditions:

- Actor is logged in

- Actor navigates to sickness/risk status page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor enters the interface for viewing the sickness/risk status.
- The individual sickness/risk status is displayed.
- Actor view it and leaves the page

Use Case: View QR Code

Participating Actor(s): User

Entry Conditions:

- Actor is logged in
- Actor navigates to viewing QR code page

Exit Conditions: Actor leaves the viewing QR code page

The Flow of Events:

- Actor enters the interface for viewing the QR code.
- The QR code that belongs to the actor is displayed which can be used for passing the HES code check at public locations and on campus easily.
- Actor leaves the viewing QR code page

Use Case: View Notification

Participating Actor(s): User

Entry Conditions: Actor navigates to page for notifications

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor opens the list of notifications in the notifications interface
- Actor can select individual notifications to see more details.

Use Case: Delete Notification

Participating Actor(s): User

Entry Conditions: Actor navigates to page for notifications

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor can delete any notification they want from the notifications screen

---

Use Case: View Individual Neighbor Information

Participating Actor(s): Admin

Entry Conditions:

- Actor is logged in
- Actor has navigated to the page for viewing everyone's neighbor information

Exit Conditions: Actor has left the page

The Flow of Events:

- Actor goes to the page for viewing neighbor information
- Actors have the ability to view every registered user's neighbors and the information of the individual neighbors

---

Use Case: View Healthy Neighbors

Participating Actor(s): Inherited from "View individual neighbor information" use case.

Entry Conditions: Inherited from "View individual neighbor information" use case.

Exit Conditions: Inherited from "View individual neighbor information" use case.

The Flow of Events:

- Actor filters the list of someone's neighbors to see the healthy ones

---

Use Case: View Risky Neighbors

Participating Actor(s): Inherited from "View individual neighbor information" use case.

Entry Conditions: Inherited from "View individual neighbor information" use case.

Exit Conditions: Inherited from “View individual neighbor information” use case.

The Flow of Events:

- Actor filters the list of someone’s neighbors to see the ones with risky status

---

Use Case: Update User’s Risk Status

Participating Actor(s): Health Center Employee

Entry Conditions:

- Actor is logged in

Exit Conditions: Actor quits the screen

The Flow of Events:

- Actor updates an individual person’s risk status depending on either a test or a risky neighbor/friend etc.
- Actor saves and leaves the screen

Use Case: Upload Physical Examination Result

Participating Actor(s): Health Center Employee

Entry Conditions:

- Actor is logged in
- Actor enters the interface for uploading physical examination results of a person

Exit Conditions: User leaves the screen

The Flow of Events:

- Actor enters the screen for uploading physical examination result on the system
- Actor locates the person from the list of registered users on the site
- Actor uploads the document
- Actor saves and leaves the screen

Use Case: Upload Vaccination Card

Participating Actor(s): User

Entry Conditions:

- Actor is logged in
- Actor enters the upload vaccination card page

Exit Conditions: User leaves the page

The Flow of Events:

- Actor enters the upload vaccination card page
- Actor can upload their vaccination card
- Actor leaves the page

Use Case: Invalid File

Participating Actor(s): User

Entry Conditions: This use extends the “Enter test results” use case. It is initiated by the system whenever an invalid vaccination card is tried to be uploaded by the user.

Exit Conditions: Actor uploads a valid vaccination card or leaves the page.

The Flow of Events: An error message is displayed.

Use Case: Enter Test Results

Participating Actor(s): Health Center Employee

Entry Conditions:

- Actor is logged in
- Actor navigates to test results page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to enter the test results page.
- A test results form is displayed.
- Actor can fill out the form with valid information and submit it to the system.
- Actor leaves the page.

Use Case: View Section Information

Participating Actor(s): Instructor

Entry Conditions:

- Actor is logged in
- Actor navigates to the section information page

Exit Conditions: Instructor leaves the page

The Flow of Events:

- Actor navigates to the section information page.
- A list of all the sections of the actor is displayed.
- Actor can select an individual section to see that section’s information.
- Once the Actor selects an individual section, the list of information of each participant from that section is displayed along with the seating plan
- Actor returns back to the list of all the sections that they teach and can select another section.
- If another section is not selected, actor leaves the screen. Otherwise, the process is repeated.

Use Case: Select Seat

Participating Actor(s): Student

Entry Conditions: Actor opens the page for selecting a seat in their sections

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor selects a seat in a particular classroom

Use Case: Make Test Appointment

Participating Actor(s): User

Entry Conditions: User navigates to screen for making test appointment

Exit Conditions: User leaves the page

The Flow of Events:

- User opens the appointment page

- User makes an appointment for a COVID test
- User leaves

Use Case: View University Wide Stats

Participating Actor(s): User

Entry Conditions: Actor navigates to the page for viewing university wide stats

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor enters the page to view university wide stats about the pandemic
- Actor views information such as the 1st dose/2nd dose percentage in the campus, total number of doses, number of applied doses (1st/2nd dose), number of tests, number of current cases etc.
- Actor quits the information page

Use Case: View Guidelines

Participating Actor(s): User

Entry Conditions: Actor navigates to the page for viewing guidelines

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor decides to view guidelines set by the university against the pandemic
- Actor is presented with a list of every guideline
- Actor closes the guidelines page

Use Case: View Weekly Report

Participating Actor(s): User

Entry Conditions: Actor opens the page for weekly report

Exit Conditions: Actor closes the page

The Flow of Events:

- Actor navigates to the page to view the weekly report

- Actor is able to see the same weekly report that is sent by the system automatically to their emails every week
- Actor closes the report

Use Case: View Announcements

Participating Actor(s): User

Entry Conditions: Actor opens the page for viewing announcements

Exit Conditions: Actor closes the page

The Flow of Events:

- Actor navigates to the announcements screen
- A list of every important announcement made by the university regarding the pandemic is displayed
- Actor closes the page

Use Case: Update University Wide Stats

Participating Actor(s): Admin

Entry Conditions: Actor navigates to the page for updating stats

Exit Conditions: Actor saves and leaves the page

The Flow of Events:

- Actor opens the page for updating university wide stats
- Actor updates stats according to the current information
- Actor leaves the page

Use Case: Create/Modify Guidelines

Participating Actor(s): Admin

Entry Conditions: Actor commences the page for creating and modifying guidelines

Exit Conditions: Actor closes the page

The Flow of Events:

- Actor opens the page

- Actor is able to create new guidelines, as well as edit the existing ones to be more inline with the pandemic conditions
- Actor saves the changes and leaves the page

Use Case: Update Announcements

Participating Actor(s): Admin

Entry Conditions: Actor enters the page for updating announcements

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor updates the important announcements made to the university members for them to keep in mind during the pandemic

---

Use Case: Create/Modify Weekly Report

Participating Actor(s): Admin

Entry Conditions: Actor opens the relevant page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to the interface for creating or modifying the weekly report
- Actor is able to create a new report if needed, or modify the weekly report created by the system to include/exclude some information

---

Use Case: Create Violation Report

Participating Actor(s): User

Entry Conditions: Actor navigates to violation report page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to the violation report page.
- A page which includes options to either create a violation report or view previously created forms are displayed.
- This use case includes View Previous Forms Created use case which allows to view previously created violation reports that can be

accessed if the actor decides to view previously created violation reports in the violation report page. If the decision is made, the list of previous violation reports are displayed and actors can select any of them and see them in more detail. After viewing that individual report, the actor can return back to the list of previous violation reports. Else, the ctor leaves the previous violation reports page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Create Request Form

Participating Actor(s): User

Entry Conditions: Actor navigates to request form page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to request form page
- A page which includes options to either create a request form or view previously created forms are displayed.
- This use case includes View previous forms created use case which allows to view previously created request forms that can be accessed if the actor decides to view previously created request forms in the request form page. If the decision is made, the list of previous request forms are displayed and actors can select any of them and see them in more detail. After viewing that individual report, the actor can return back to the list of previous request forms. Else, the actor leaves the previous request forms page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Create Feedback Form

Participating Actor(s): User

Entry Conditions: Actor navigates to feedback form page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to feedback form page
- A page which includes options to either create a feedback form or view previously created forms are displayed.

- This use case includes View previous forms created use case which allows to view previously created feedback forms that can be accessed if the actor decides to view previously created feedback forms in the feedback form page. If the decision is made, the list of previous feedback forms are displayed and users can select any of them and see them in more detail. After viewing that individual report, the actor can return back to the list of previous feedback forms. Else, the actor leaves the previous feedback forms page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Handle Violation Reports

Participating Actor(s): Admin

Entry Conditions: Actor navigates to violation reports page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to violation reports page
- A page which includes options to either view all violation reports or delete previously created violation reports that were created by students, instructors and bilkent security are displayed.
- This use case includes the View all use case which allows to view all violation reports that can be accessed if the actor decides to view all violation reports in the violation reports page. If the decision is made, the list of all created violation reports are displayed. Actors select an individual report to see that report in more detail or leave the violation reports page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Handle Feedback Forms

Participating Actor(s): Admin

Entry Conditions: Actor navigates to feedback forms page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to feedback forms page

- A page which includes options to either view all feedback forms or delete previously created feedback forms that were created by students, instructors and bilkent security are displayed.
- This use case includes the View all use case which allows to view all feedback forms that can be accessed if the actor decides to view all feedback forms in the feedback forms page. If the decision is made, the list of all created feedback forms are displayed. Actors select an individual feedback form to see that form in more detail or leave the feedback forms page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Handle Request Forms

Participating Actor(s): Admin

Entry Conditions: Actor navigates to request forms page

Exit Conditions: Actor leaves the page

The Flow of Events:

- Actor navigates to request forms page
- A page which includes options to either view all request forms or delete previously created request forms that were created by students, instructors and bilkent security are displayed.
- This use case includes the View all use case which allows to view all request forms that can be accessed if the actor decides to view all request forms in the request forms page. If the decision is made, the list of all created request forms are displayed. Actors select an individual request form to see that form in more detail or leave the request forms page.
- Actor leaves the page and returns back to the Report and Request page.

Use Case: Send Friend Request

Participating Actor(s): Student

Entry Conditions: Actor decides to add a friend and enters the relevant screen

Exit Conditions: Actor leaves the friend request screen

The Flow of Events:

- Actor wants to add a friends and opens the interface
- Actor enters required information to add a friend and sends a friend request
- Actor leaves the screen

Use Case: No Such User

Participating Actor(s): Student

Entry Conditions: This use case extends the “Send friend request” use case. It is initiated by the system whenever the user enters invalid information in the friend request interface.

Exit Conditions: Actor enters valid information

The Flow of Events:

- Actor supplies invalid information in the friend request screen
- Actor either leaves the interface or supplies correct information

---

Use Case: Remove Friend

Participating Actor(s): Student

Entry Conditions: Actor navigates to the screen for removing friends

Exit Conditions: Actor leaves the screen

The Flow of Events:

- Actor opens the screen for removing a friend
- Actor selects the friend they want to remove from their friends list
- Actor leaves the page

Use Case: Accept/Decline Friend Request

Participating Actor(s): Student

Entry Conditions: Actor opens the screen for accepting or declining friend requests

Exit Conditions: Actor leaves the screen after they are done

The Flow of Events:

- Actor opens the screen for incoming friend requests

- Actor chooses to either accept or decline the friend requests they have received
- Actor leaves the screen

Use Case: View Friend Status

Participating Actor(s): Student

Entry Conditions: Actor enters the screen for viewing their friends' statuses

Exit Conditions: Actor leaves the screen

The Flow of Events:

- Actor opens the interface for viewing their friends' statuses
- Actor is presented with a list of their friends, as well as the statuses of each friend
- Actor leaves the screen

Use Case: View Risky Friends

Participating Actor(s): Inherited from "View friend status" use case.

Entry Conditions: Inherited from "View friend status" use case.

Exit Conditions: Inherited from "View friend status" use case.

The Flow of Events:

- In the interface for viewing friend statuses, the Actor filters out their friends who are in a risky (contacted/sick) status, sickness wise.

Use Case: View Healthy Friends

Participating Actor(s): Inherited from "View friend status" use case.

Entry Conditions: Inherited from "View friend status" use case.

Exit Conditions: Inherited from "View friend status" use case.

The Flow of Events:

In the interface for viewing friend statuses, the Actor filters out their friends who are in healthy status, sickness wise.

Class Diagram

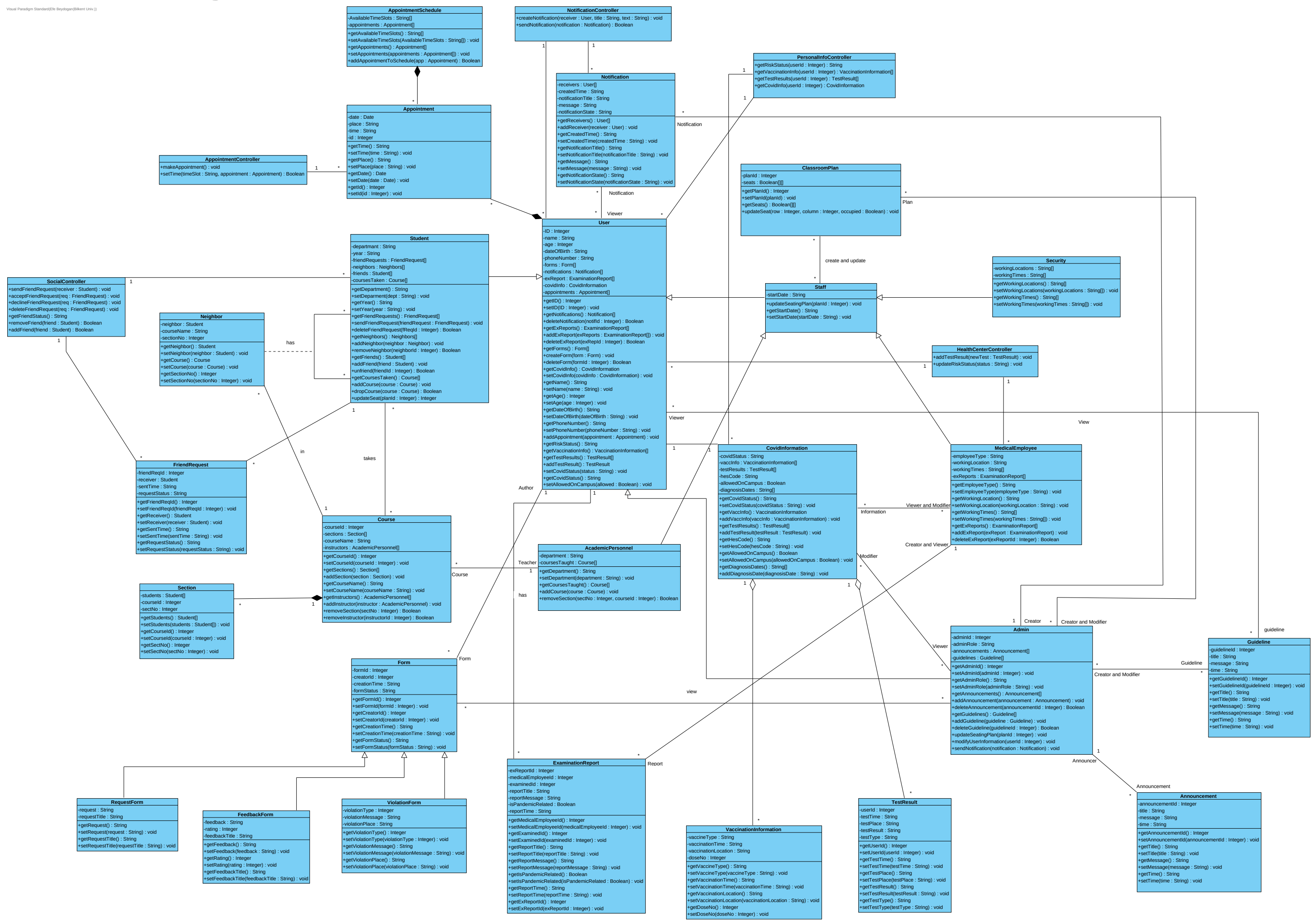


Fig. 4: Class Diagram

The class diagram consists of 30 classes. The diagram presents the fundamental objects we will need when implementing the project.

User class: The User class is the superclass of Student and Staff classes. It includes properties and methods that are common for every user except admins. These information include the ID, name, age, date of birth and covid information of the user etc.

Student class: The Student class is the child of the User class. It includes properties and methods that every student has. It includes information such as the department, courses, neighbors and friends of the student.

Staff class: The Staff class is the child of the User class. It is the parent of AcademicPersonnel, Security and MedicalEmployee classes. It includes the functionality to update a classroom seating plan, which is common for every Bilkent staff member.

AcademicPersonnel class: AcademicPersonnel class is the child of the Staff class. It contains the department information of the instructor as well as functionality to add and remove courses that are taught by the personnel.

Security class: Security class is the child of Staff class. It has the working locations and working times of the security personnel. Also, when creating a form, forms created by Security members are intended to have higher priorities.

MedicalEmployee class: MedicalEmployee class is the subclass of the Staff class. It contains the examination reports created by the health center employees.

Admin class: Admin class represents the administrators of the system. It has the ability to access and modify every information in the system.

Neighbor class: The Neighbor class is the association class which represents a many to many relationship between students. Every student has many neighbors and the neighbor class holds information about the name, course and section of the neighbor of a student.

FriendRequest class: The FriendRequest class has a 1 to many relationship with the Student class. Every student can receive many friend requests, but a particular friend request can be sent by only one student. The FriendRequest class holds information including an ID, the receiver and the time the request was sent.

Course class: The Course class has a many to many relationship with the Student class, every student can take many courses and a course can have many students in it. Course class also has a 1 to many relationship with the Neighbor class. The neighborship can only be present in one course section, whereas a course can have many neighbor relations.

Section class: The Section class has a 1 to many relationship with the Course class. One course can have many courses, and a section can only belong to one Course. Also, there is a composition relationship between the Section and Course classes. A Section can't exist without a Course.

Form class: The Form class is the superclass of RequestForm, FeedbackForm and ViolationForm classes. It has a 1 to many relationship with the User class. A User can have many forms while a particular form can only belong to one User. The Form class contains information that is common to all Form types, such as the form id, creator id and the creation time.

RequestForm class: RequestForm class is the subclass of Form class. Request forms are sent to the university for pandemic related requests. It contains the request title and request body.

FeedbackForm class: FeedbackForm class is the subclass of the Form class. Feedback forms are sent to the maintainers of our app to improve user experience. It contains the feedback title and body, as well as a rating.

ViolationForm class: ViolationForm class is the subclass of the Form class. Violation forms are sent by any member of the university for review when they encounter a violation of pandemic guidelines. The object contains the violation type, a message and the place the violation took place.

ExaminationReport class: Examination reports are created by medical employees and assigned to users. There is a 1 to many relationship between the ExaminationReport class and the User and MedicalEmployee classes. A User can have many examination reports and a report can only belong to one user. A medical employee can create many reports but a report can only be created by one. The class contains the examination report id, the employee that created it, as well as a report body and report time.

Notification class: Notifications are sent to all users regarding urgent information. Notifications are created and sent by admins and received by the users.

ClassroomPlan class: The ClassroomPlan class holds information about the seating plan of a classroom. It can be created and modified by administrators and staff members.

CovidInformation class: CovidInformation class holds all information of a User related to the pandemic. It holds the covid status, HES code, vaccination information and previous test results of a User.

VaccinationInformation class: VaccinationInformation class holds information about the vaccination dates, types and the number of doses. It has an aggregation relation with the CovidInformation class.

TestResult class: TestResult class holds information about a test result, which consists of a user id, test type, time, location etc. It has an aggregation relation with the CovidInformation class.

Announcement class: Announcement class holds information about an announcement which includes the title and the announcement message. These announcements are created by admins. An admin can create many announcements but, an announcement can only be created by one admin.

Guideline class: Guideline class holds information about a guideline related to the pandemic. It can be created by admins and viewed by all users. The instances of the guideline class will be displayed on a page that is made for the pandemic guidelines. People that don't obey these guidelines can be reported to the administration through the violation reports.

Appointment class: Appointment class holds the information of an appointment which has a date, place and time. It has an aggregation relation with the classes User and AppointmentSchedule.

AppointmentSchedule class: AppointmentSchedule has two attributes which are available time slots and appointments. This class can get and set available time slots and existing appointments. It can also add appointments to the schedule.

AppointmentController class: AppointmentController class can create an empty appointment object with a unique appointmentId, which will be filled with the information that the user will enter (the time slot he picks etc.). Additionally, it can set the time in a currently created appointment, and the reason why this method exists is because a boundary object that the user interacts with can only communicate with a controller object, and this method of the controller will call the setTime method of the Appointment class.

SocialController class: SocialController class can send a friend request, it can accept or decline the given friend request. It can delete the given friend request and it can view the status of a friend. Additionally, it can add a new friend and remove an existing friend.

PersonalInfoController class: PersonalInfoController class can retrieve a user’s risk status, vaccination information, existing covid test results and complete covid information by using the user id.

NotificationController class: NotificationController class can create a Notification with the information provided by the system or admin and it can send the created notification to the given receiver/s.

HealthCenterController class: HealthCenterController class can add a new test result whose information will be given by the Medical Employee and it can update the risk status of a user according to the new test result as well as the user’s neighbors.

Dynamic Models

Form Management Activity Diagram

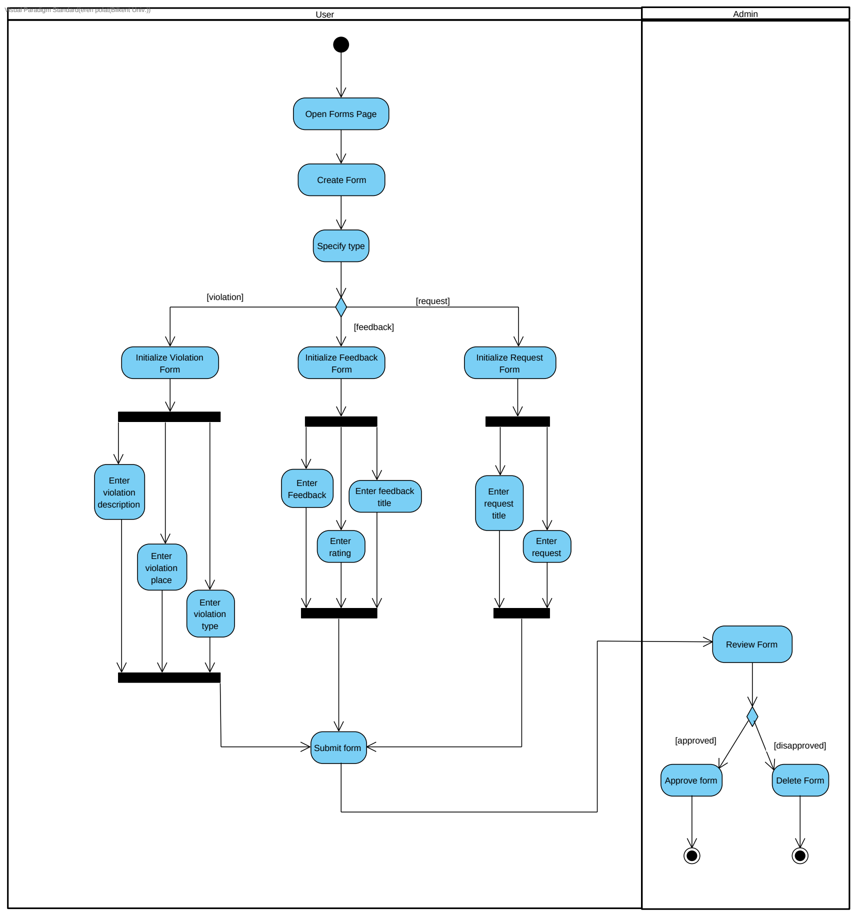


Fig. 5: Form Management Activity Diagram

Scenario: User opens the form page and creates a blank form, where he first specifies its type. After specifying the form type; he initializes the form with necessary information. If it is a violation form, he enters violation description, violation place, and violation type. If it is a feedback form; he enters his feedback, rating, and feedback title. If it is a request form; he enters request title and the request. Bars are used to depict that all of these necessary information should be entered before continuing to submit the form, but they don't necessarily have to be in a particular order (for example, one can first enter violation type and then violation place; or the other way around). After entering necessary information, the user submits the form which is reviewed by an admin, where it is either approved or disapproved (in this case it is deleted).

Authentication Activity Diagram

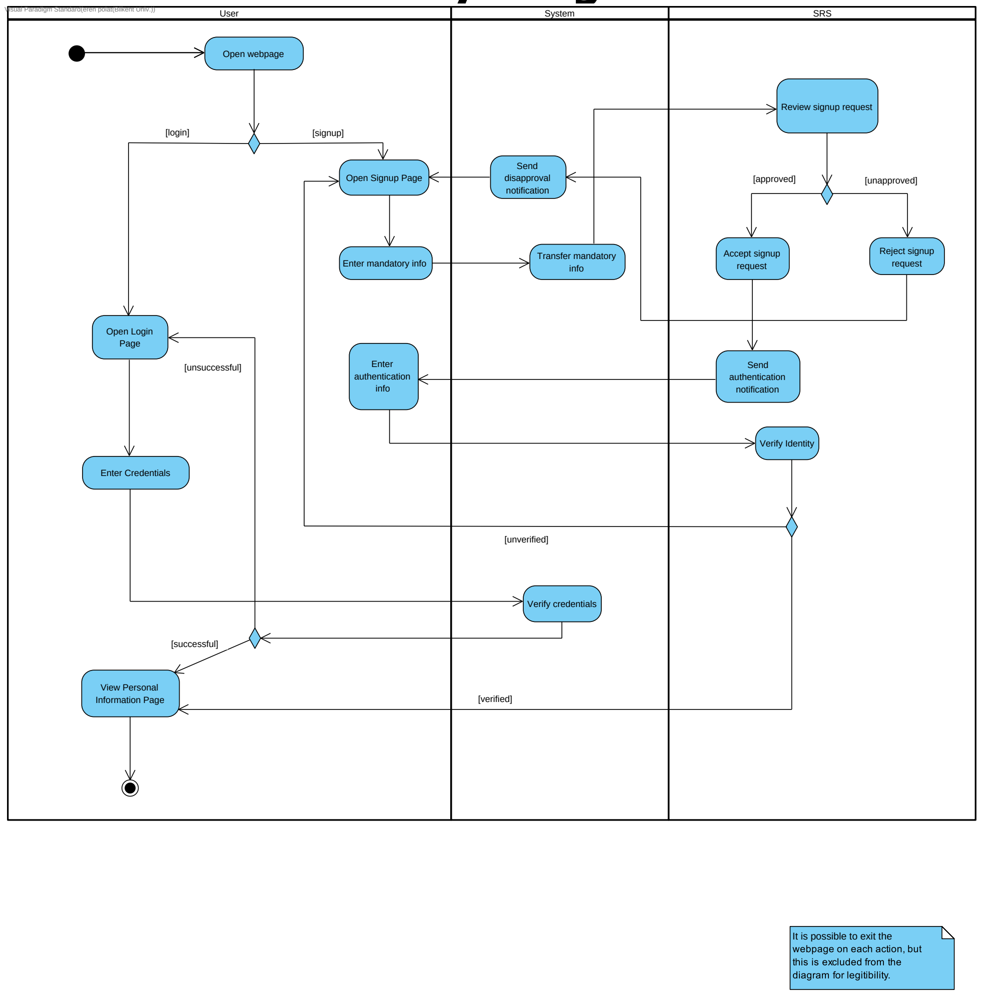


Fig. 6: Authentication Activity Diagram

Scenario: User opens the webpage, where he picks whether he wants to login with a preexisting account or sign-up to create an account if hasn't registered yet. Case 1, if he chooses to login, he opens the login page and enters his credentials. If credentials match, login is successful and the user is navigated to personal information page. If the credentials do not match, the user remains in the login page. Case 2, if he chooses to sign up, the user enters mandatory information -which is transferred to SRS- for verification. If information matches with the ones in SRS, sign-up request is accepted. For two-factor authentication, the SRS sends an authentication notification to the user and user enters information required for authentication. If info matches, SRS verifies identity, and the user is navigated to personal information page. If authentication information does not match, user is directed back to sign-up page

Update Seat Activity Diagram

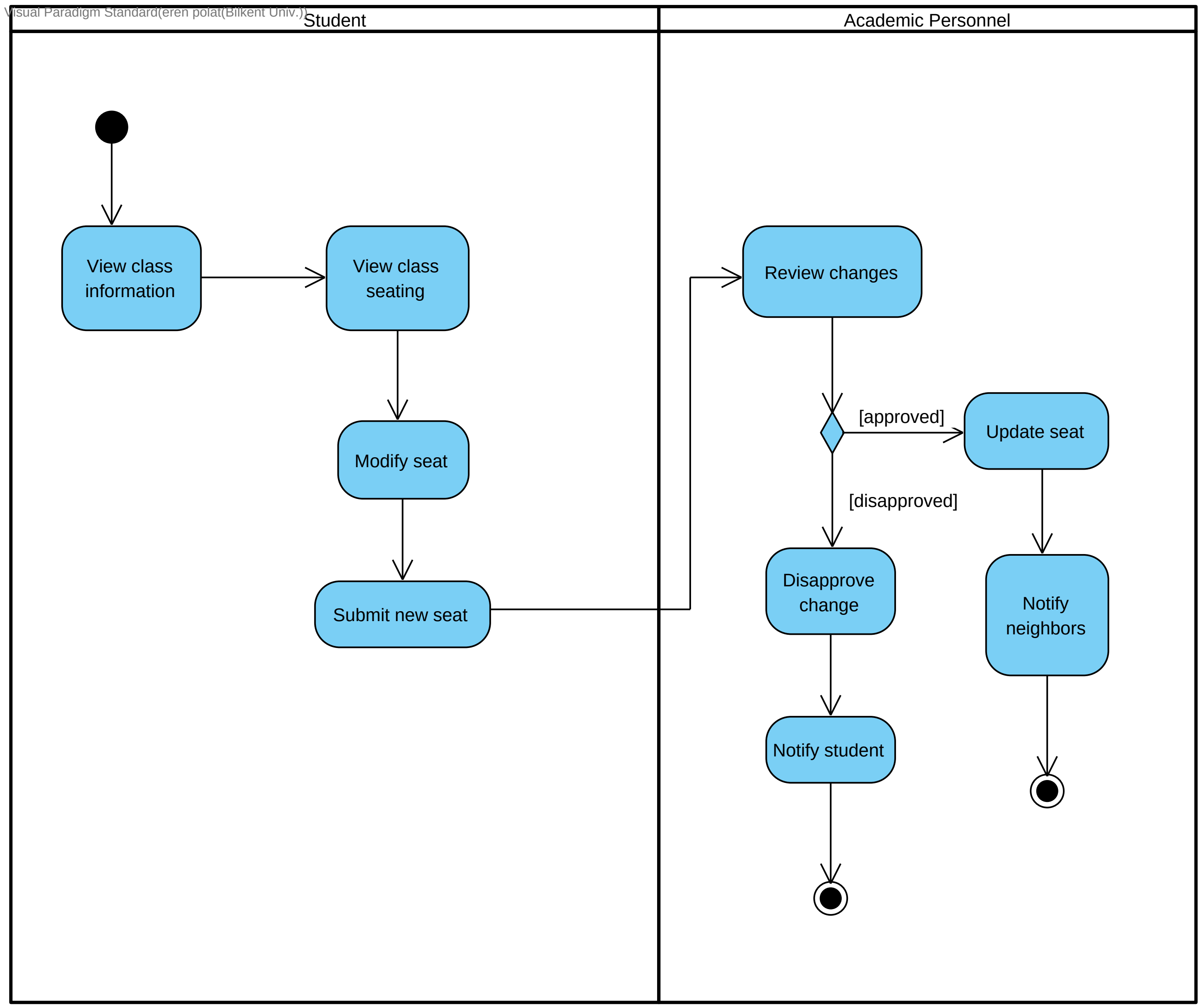


Fig. 7: Update Seat Activity Diagram

Scenario: Student opens the class information page and then navigates to the seating plan of the class. Student changes his/her seat by selecting an available seat from the seating plan of the classroom. After the modification, the student submits his/her new seat. Then, the academic personnel reviews this change. If the academic personnel approves the change in the seating plan, the seating plan is updated and the neighbors are notified. If the academic personnel disapproves of the change, then she/he notifies the student.

Sequence Diagram for Make Appointment Use Case

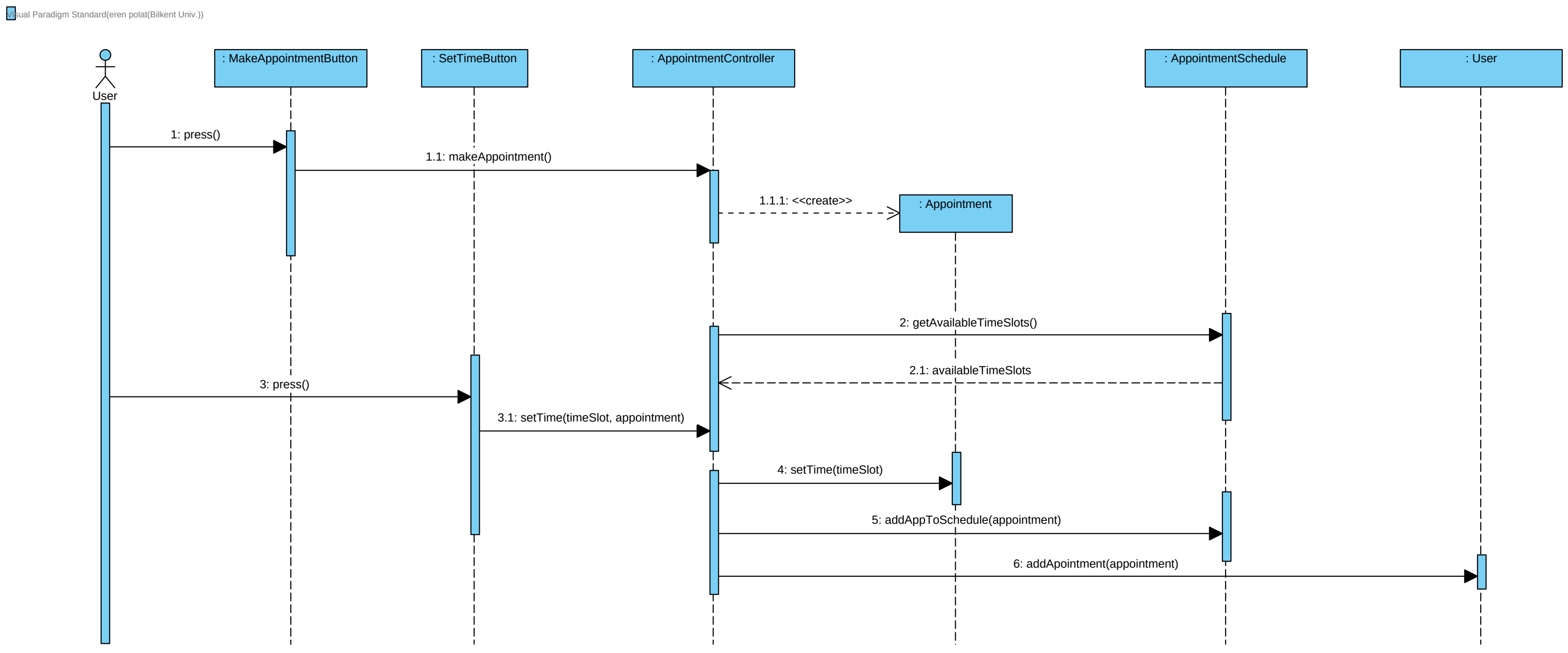


Fig. 8: Sequence Diagram for Make Appointment Use Case

Scenario: User navigates to the screen of making an appointment and clicks the MakeAppointmentButton to create an appointment instance (the boundary object interacts with the controller, which creates an appointment object). Then, the available time slots are fetched from the schedule and user picks an available time. The appointment object is assigned to this time slot, and this time slot becomes reserved in the appointment schedule.

Sequence Diagram for Send Notification Use Case

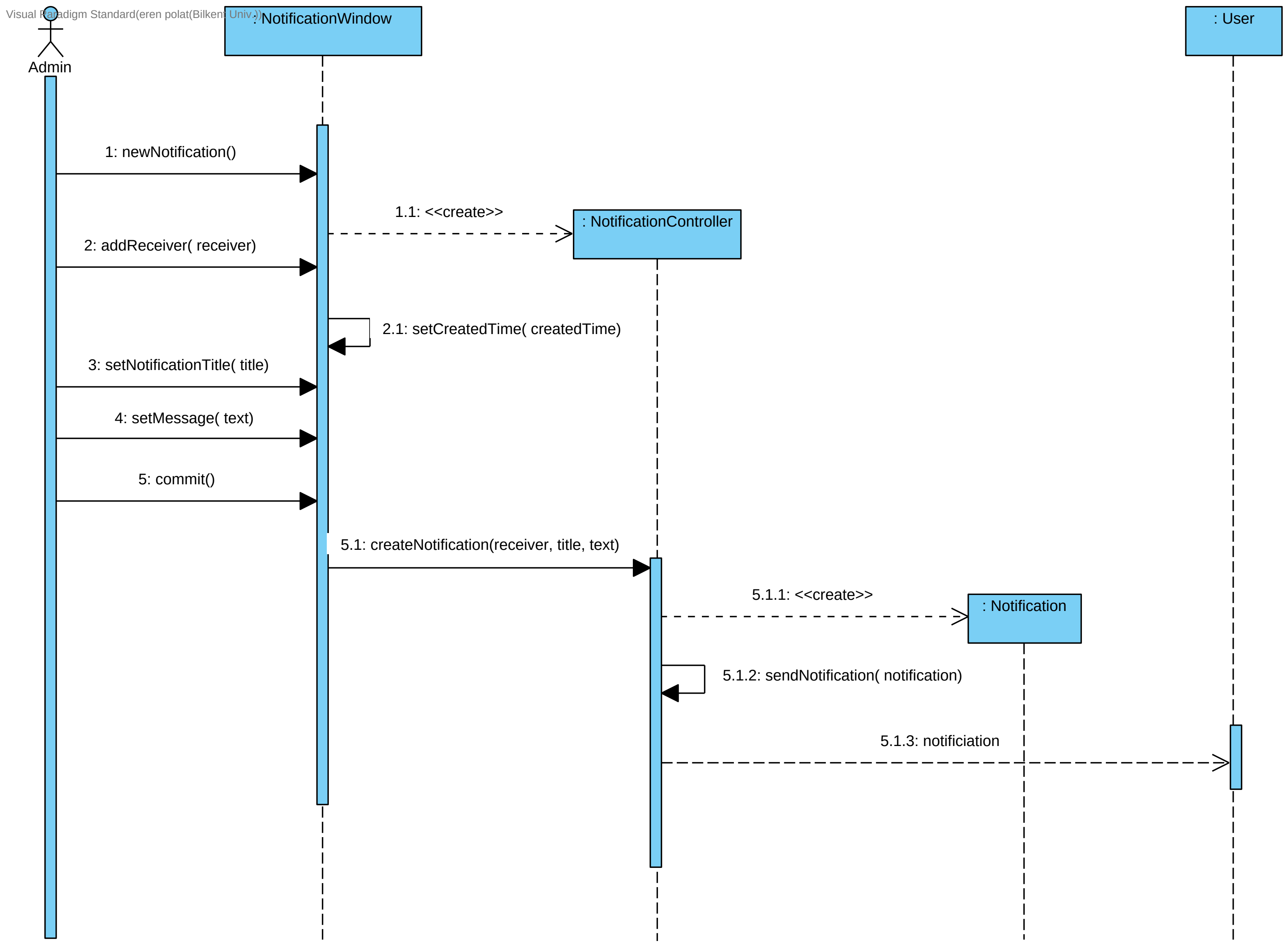

Visual Paradigm Standard(eren polat(Bilkent Univ.))
Admin
: NotificationWindow
: NotificationController
: Notification
: User
1: newNotification()
1.1: <<create>>
2: addReceiver(receiver)
2.1: setCreatedTime(createdTime)
3: setNotificationTitle(title)
4: setMessage(text)
5: commit()
5.1: createNotification(receiver, title, text)
5.1.1: <<create>>
5.1.2: sendNotification(notification)
5.1.3: notificiation

Fig. 9: Sequence Diagram for Send Notification Use Case

Scenario: An admin navigates to the notification window and creates a blank notification form to be filled. He adds the receivers of the notification, sets the title and writes his message. He then commits the notification, where the information he entered is used to create a new notification; which is sent to designated receivers.

Sequence Diagram for Update User’s Risk Status Use Case

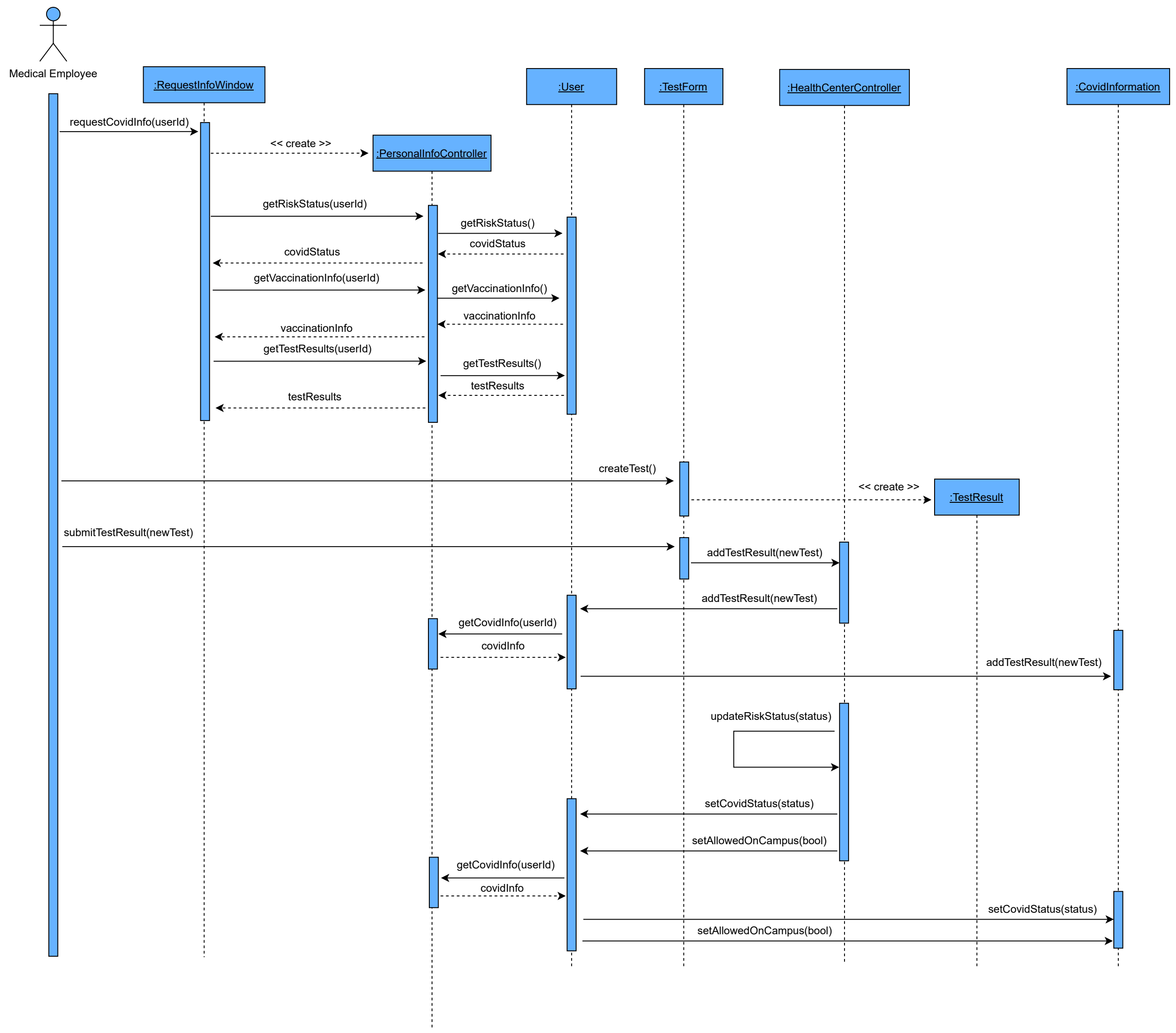


Fig. 10: Sequence Diagram for Update User’s Risk Status Use Case

Scenario: A medical employee navigates to the window where s/he can view a user’s information by entering the id of the user. When the information is requested, the PersonalInfoController becomes activated and retrieves the user’s related information. After the information retrieval, the medical employee opens the test submission place (TestForm which is a boundary object) and submits the user’s new test result. Then the system automatically updates the user’s risk status and decides whether that user is allowed on campus or not.

Sequence Diagram for Send Friend Request Use Case

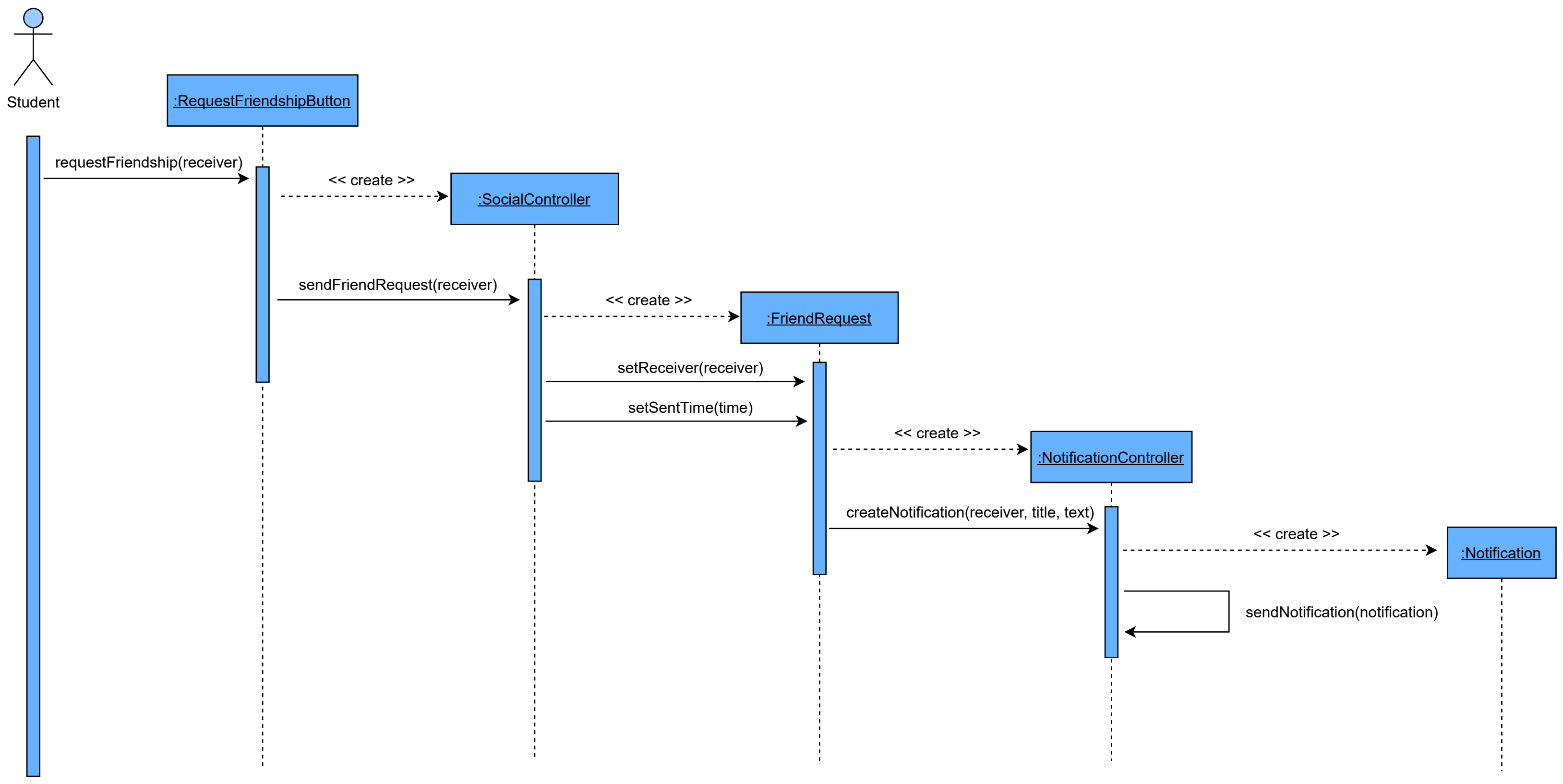


Fig. 11: Sequence Diagram for Send Friend Request Use Case

Scenario: A student clicks the RequestFriendshipButton to become a friend of a particular user. This action of student's activates the SocialController and the system creates a FriendRequest object and fills the necessary information such as receiver and time. Then the FriendRequest object triggers the NotificationController, and the controller sends a notification to the receiver about the request.

State Diagram for User’s Health

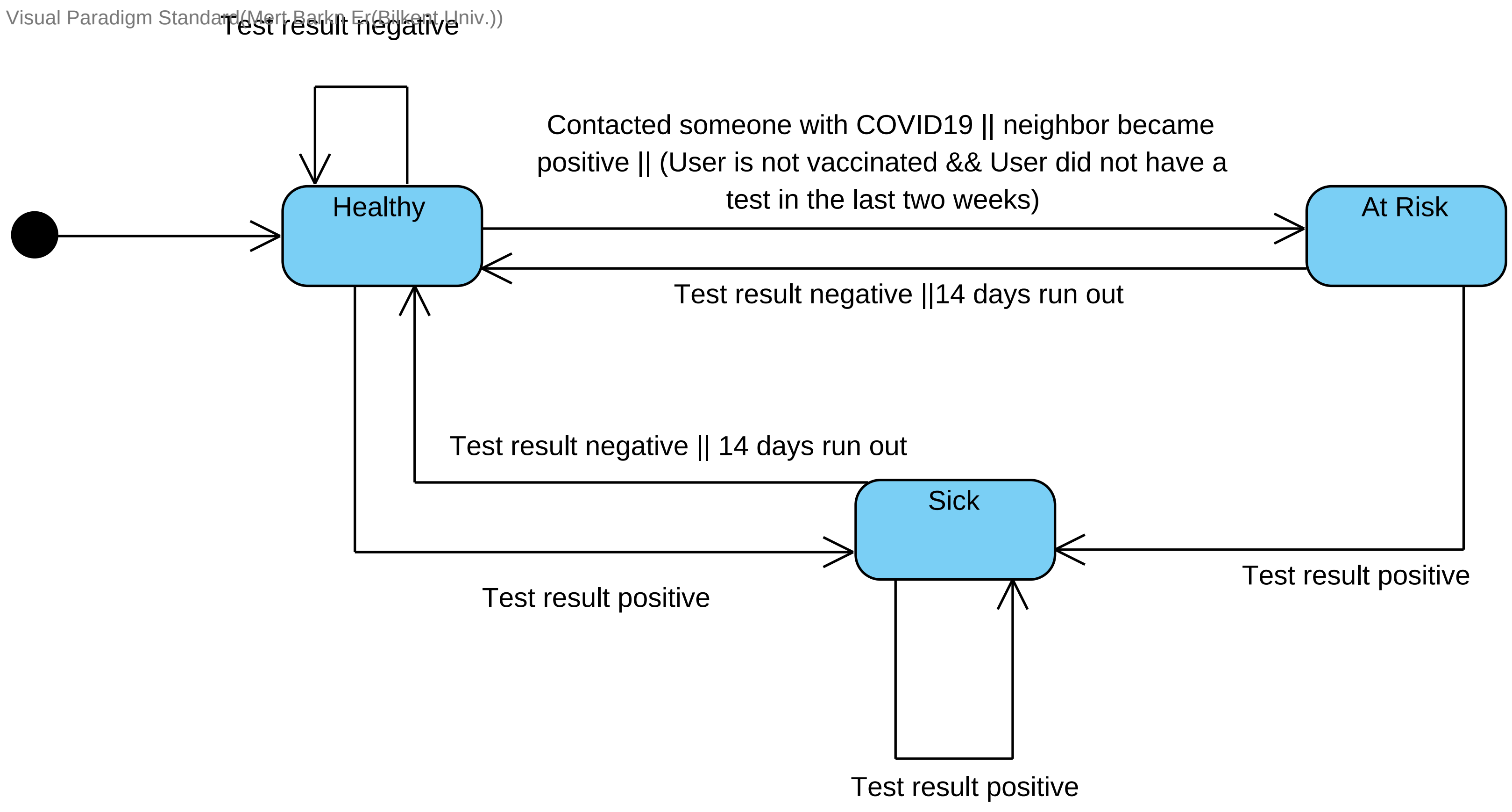

Visual Paradigm Standard(Mert Barkın Er(Bilkent Univ.))
Test result negative
Contacted someone with COVID19 || neighbor became positive || (User is not vaccinated && User did not have a test in the last two weeks)
Healthy
At Risk
Test result negative ||14 days run out
Test result negative || 14 days run out
Sick
Test result positive
Test result positive
Test result positive

Fig. 12: State Diagram for User’s Health

A user’s default health state is Healthy. As long as the user’s test results are negative the health state also stays as Healthy. If a user’s test result comes back as positive the health state is set as Sick. If a user is healthy and contacted someone with COVID19, if one of their neighbors became positive, or if they did not have a test in two weeks when they did not have the vaccine their health state becomes At Risk. If an At Risk user quarantines for 2 weeks or has a test that produces a negative result their health state is set as Healthy. If an At Risk user has a test that produces a positive result their health state is set as Sick. A Sick user stays as Sick as long as their test results come back positive. If a Sick user’s test result comes back as negative their health state is set to Healthy.

State Diagram for Form

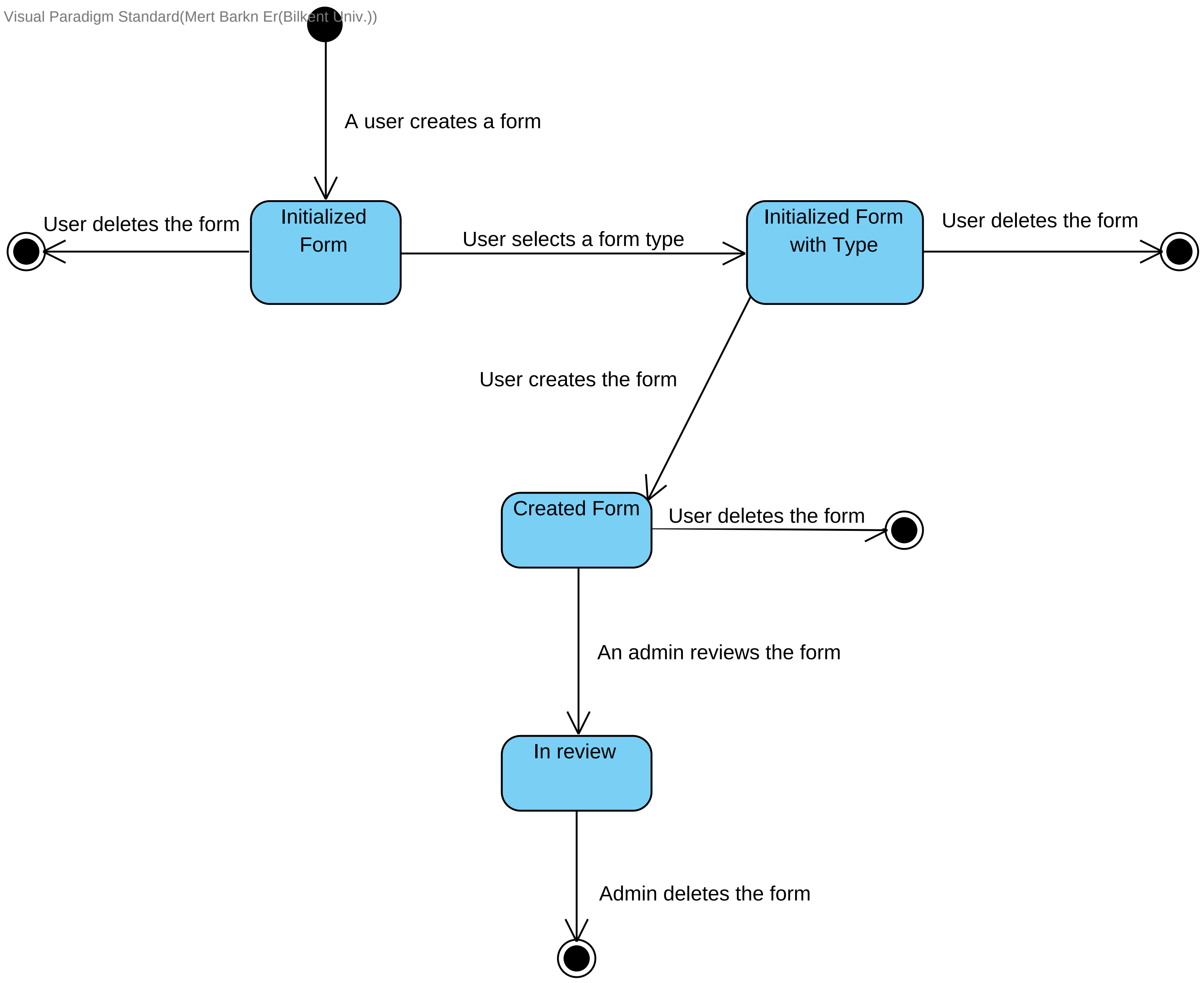


Fig. 13: State Diagram for Form

A user creates a form, which is set as an Initialized Form. After that user selects a form type (e.g. a request form) the form’s state becomes Initialized Form with Type. After a user fills the form they chose, the form’s state becomes Created Form. When an admin decides to review this form its state is set as In review. At any point either the user who created this form or an admin can delete this form.

State Diagram for Seating Plan

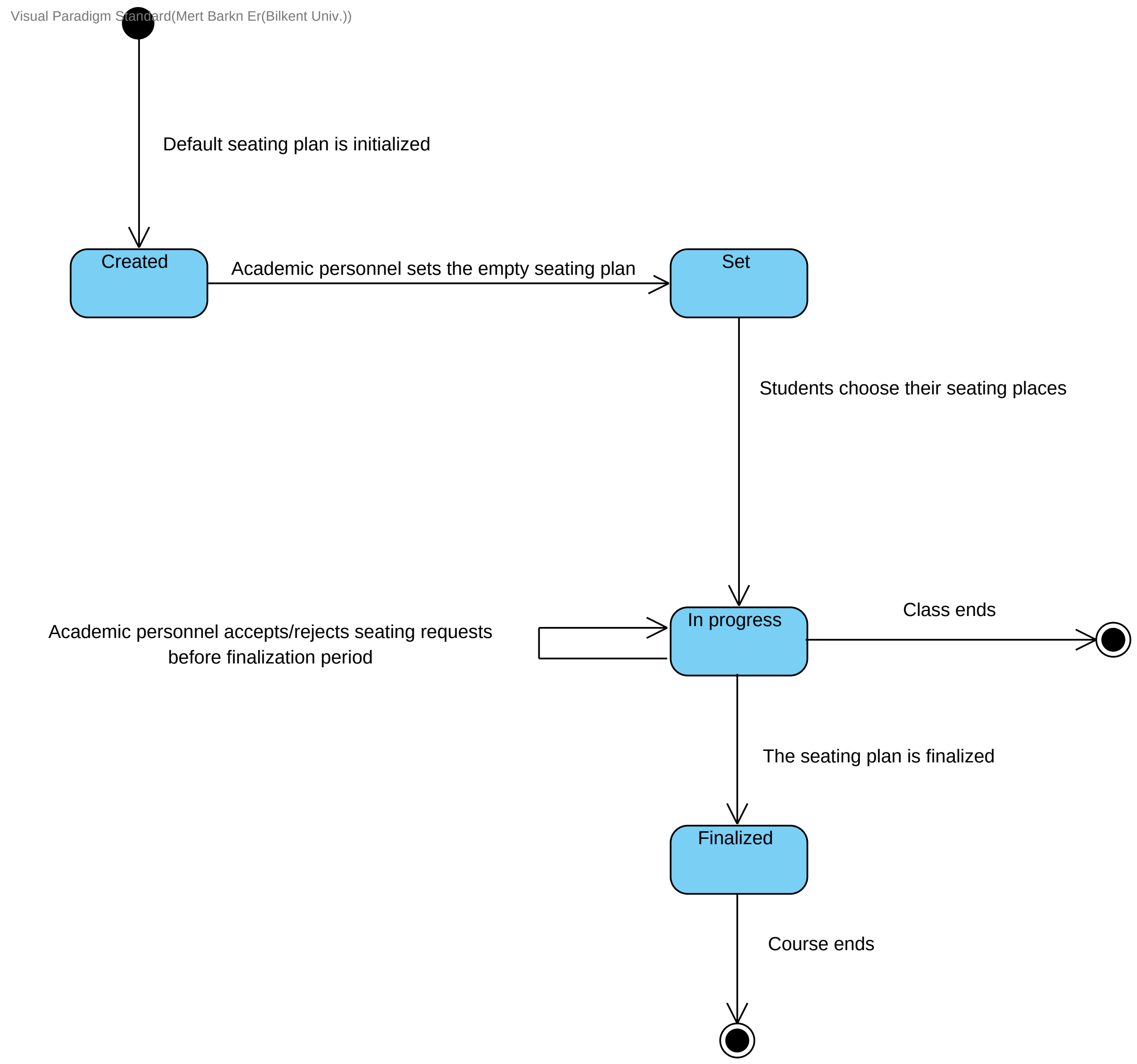
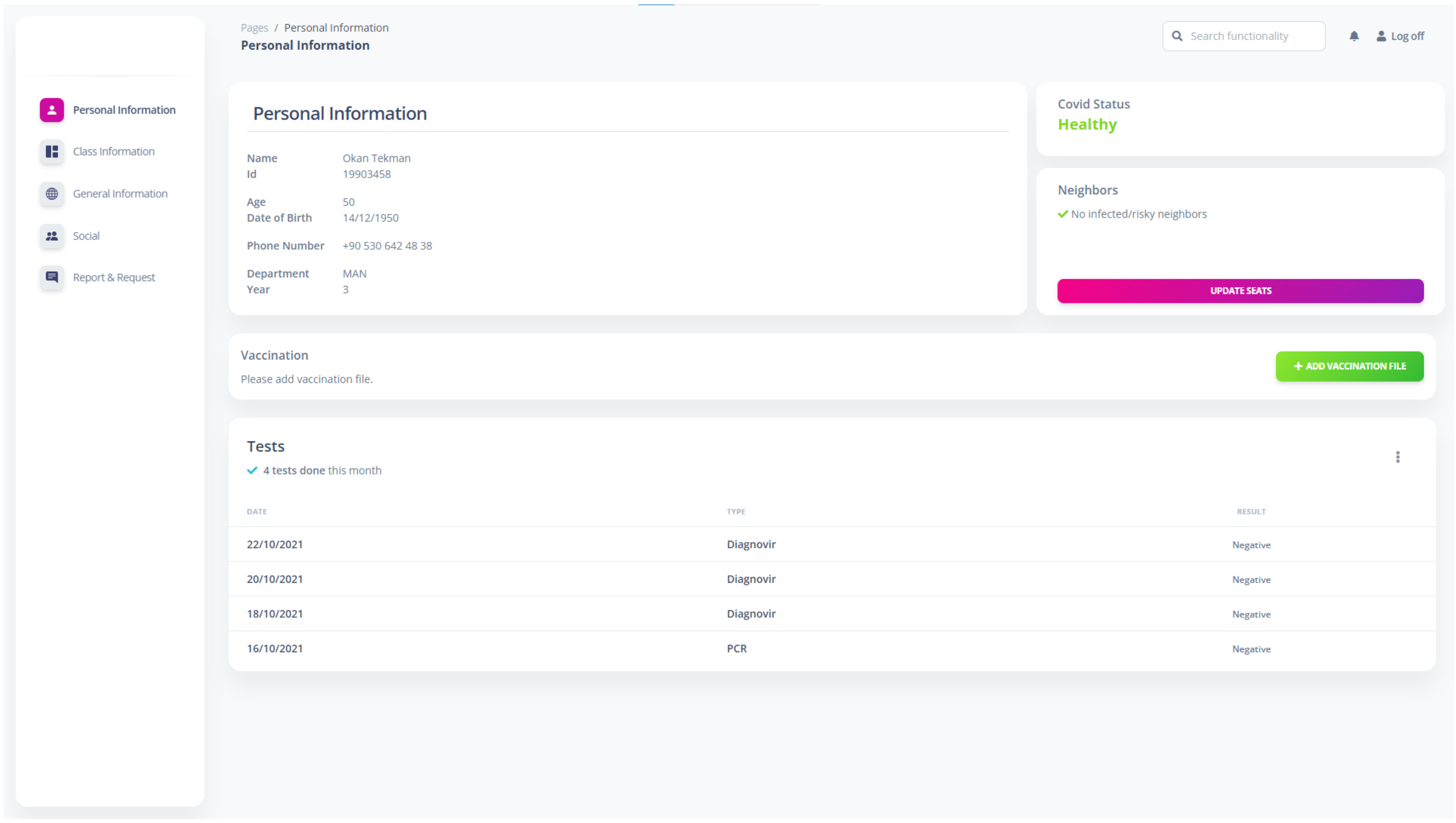

Visual Paradigm Standard(Mert Barkn Er(Bilkent Univ.))
Default seating plan is initialized
Created
Academic personnel sets the empty seating plan
Set
Students choose their seating places
Academic personnel accepts/rejects seating requests before finalization period
In progress
Class ends
The seating plan is finalized
Finalized
Course ends

Fig. 14: State Diagram for Seating Plan

When a section is initialized, a seating plan for it is also created, with its state being Created. The academic personnel for that section modifies this seating plan and creates an empty grid-like seating plan (e.g. a 6 x 20 seating plan) and the seating plans state becomes Set. After the seating plan becomes Set, students pick their preferred seating choices, at which point the seating plan becomes In Progress. During In Progress state, students can send new seating requests to the academic personnel who can accept/reject these requests. The seating plan becomes finalized when the finalization period ends, and its state becomes Finalized.

User Interface - Screen Mock-ups

Personal Information Page


Pages / Personal Information

Personal Information

Search functionality

Log off

- Personal Information
- Class Information
- General Information
- Social
- Report & Request

Personal Information

Name	Okan Tekman
Id	19903458
Age	50
Date of Birth	14/12/1950
Phone Number	+90 530 642 48 38
Department	MAN
Year	3

Covid Status

Healthy

Neighbors

✓ No infected/risky neighbors

UPDATE SEATS

Vaccination

Please add vaccination file.

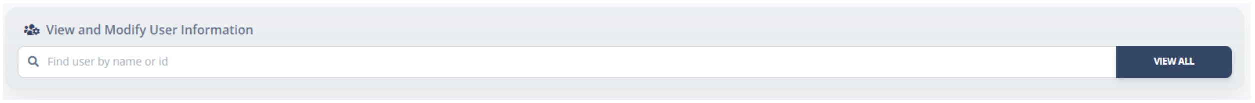
+ ADD VACCINATION FILE

Tests

✓ 4 tests done this month

DATE	TYPE	RESULT
22/10/2021	Diagnovir	Negative
20/10/2021	Diagnovir	Negative
18/10/2021	Diagnovir	Negative
16/10/2021	PCR	Negative

Fig. 15.1: Personal Information Page Mock-up

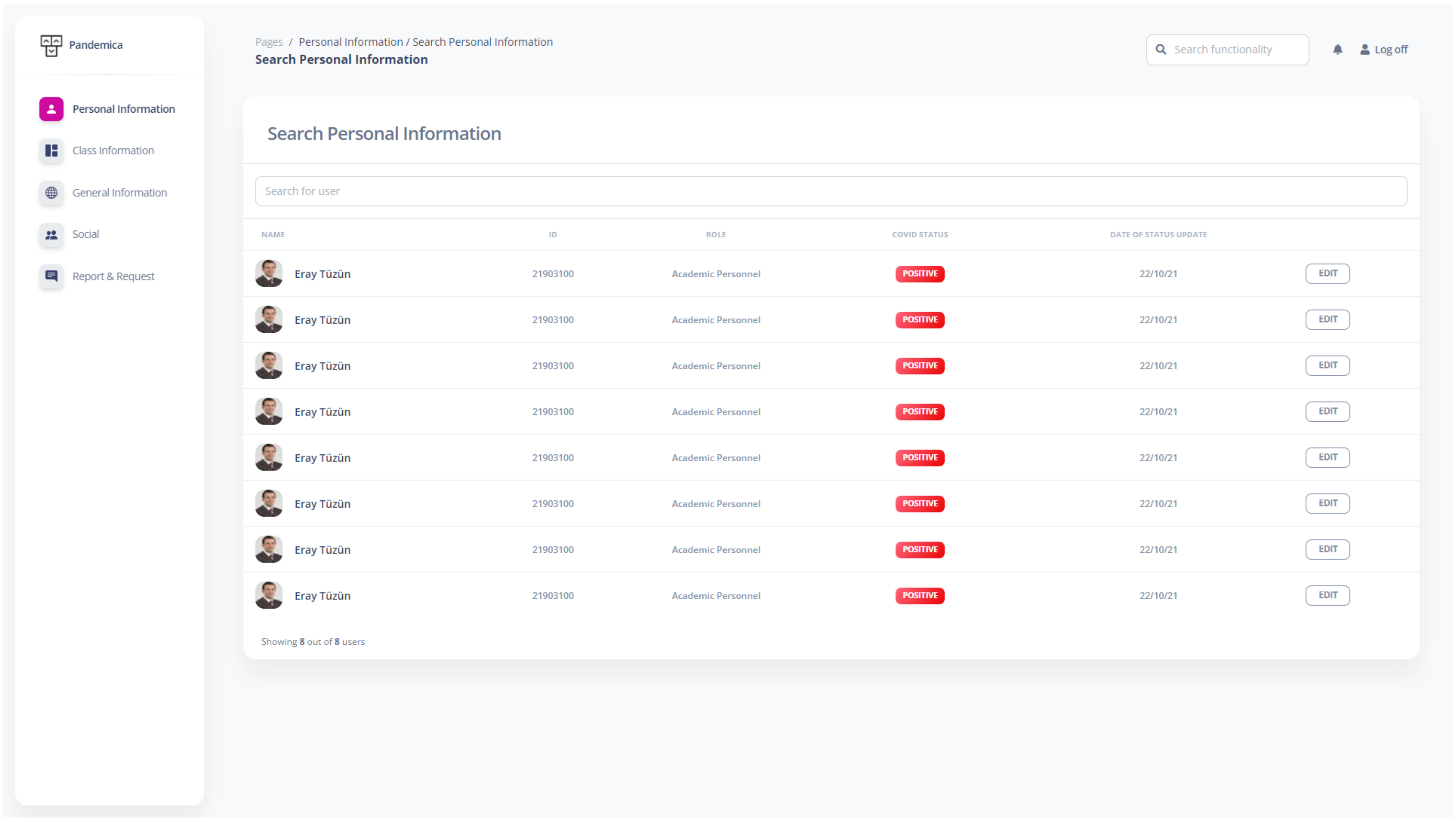

View and Modify User Information

Find user by name or id

VIEW ALL

Fig. 15.2: Admin/Health Center Employee Search Bar for Users Mock-up

Search Personal Information Page



Pandemica

Personal Information

Class Information

General Information

Social

Report & Request

Pages / Personal Information / Search Personal Information

Search Personal Information

Search functionality

Log off

Search Personal Information

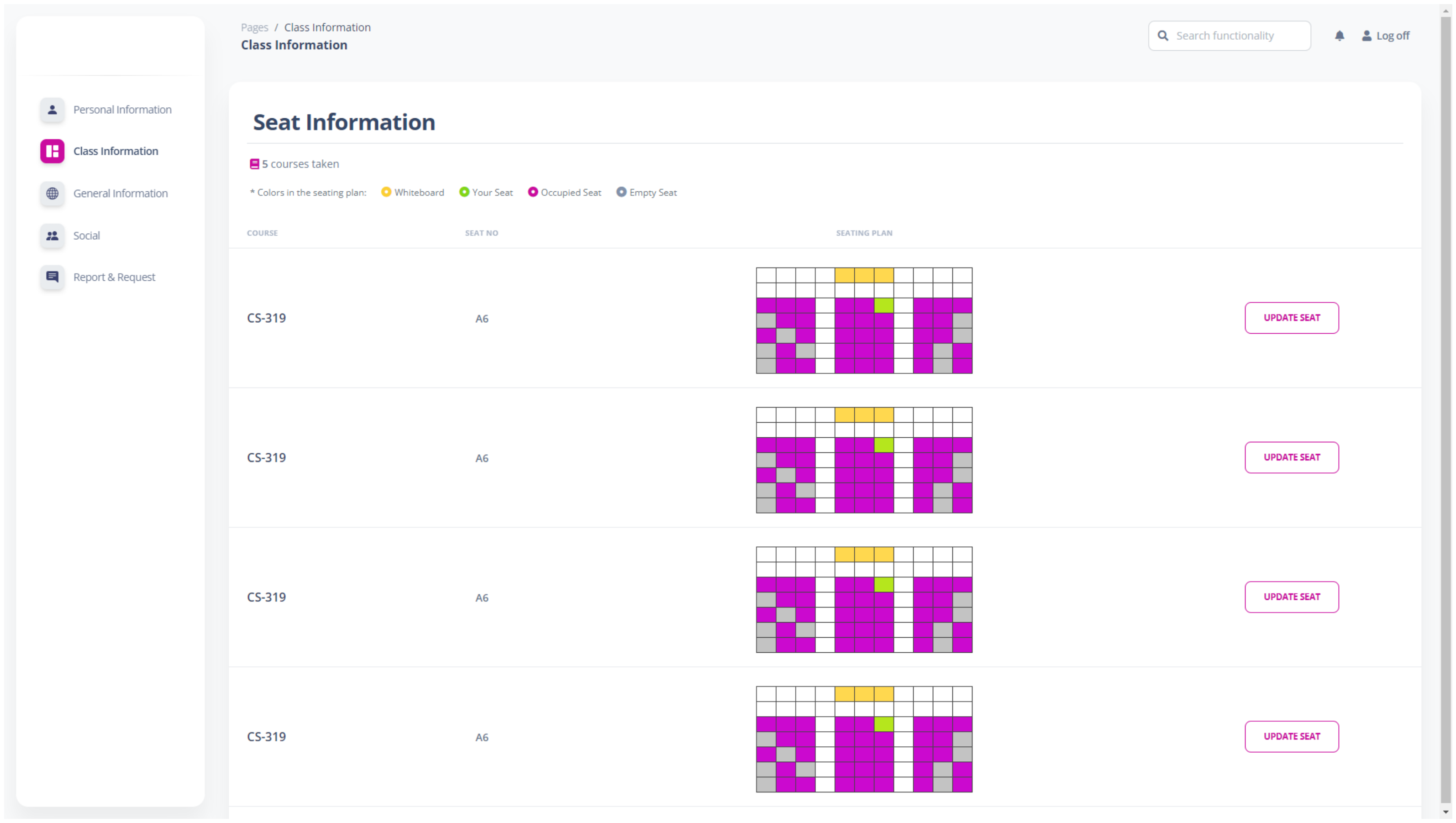
Search for user

NAME	ID	ROLE	COVID STATUS	DATE OF STATUS UPDATE	
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT
Eray Tüzün	21903100	Academic Personnel	POSITIVE	22/10/21	EDIT

Showing **8** out of **8** users

Fig. 15.3: Search Personal Information Page Mock-up

Class Information Page


Pages / Class Information

Class Information

Seat Information

5 courses taken

* Colors in the seating plan: ● Whiteboard ● Your Seat ● Occupied Seat ● Empty Seat

COURSE	SEAT NO	SEATING PLAN	
CS-319	A6		UPDATE SEAT
CS-319	A6		UPDATE SEAT
CS-319	A6		UPDATE SEAT
CS-319	A6		UPDATE SEAT

Fig. 16.1: Class Information Page Mock-up

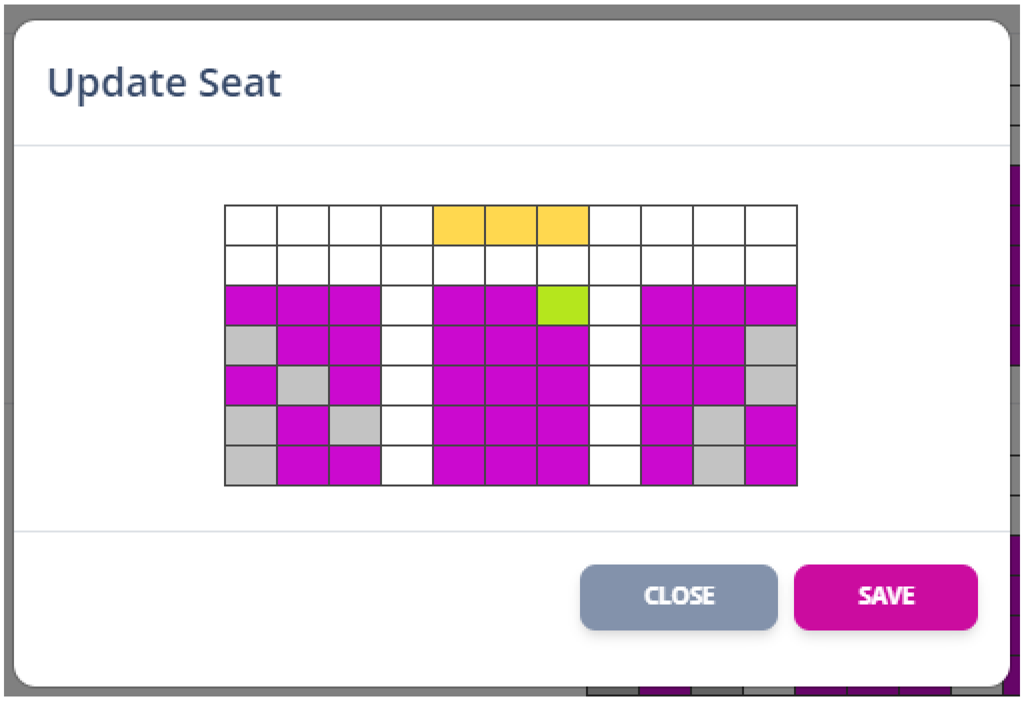
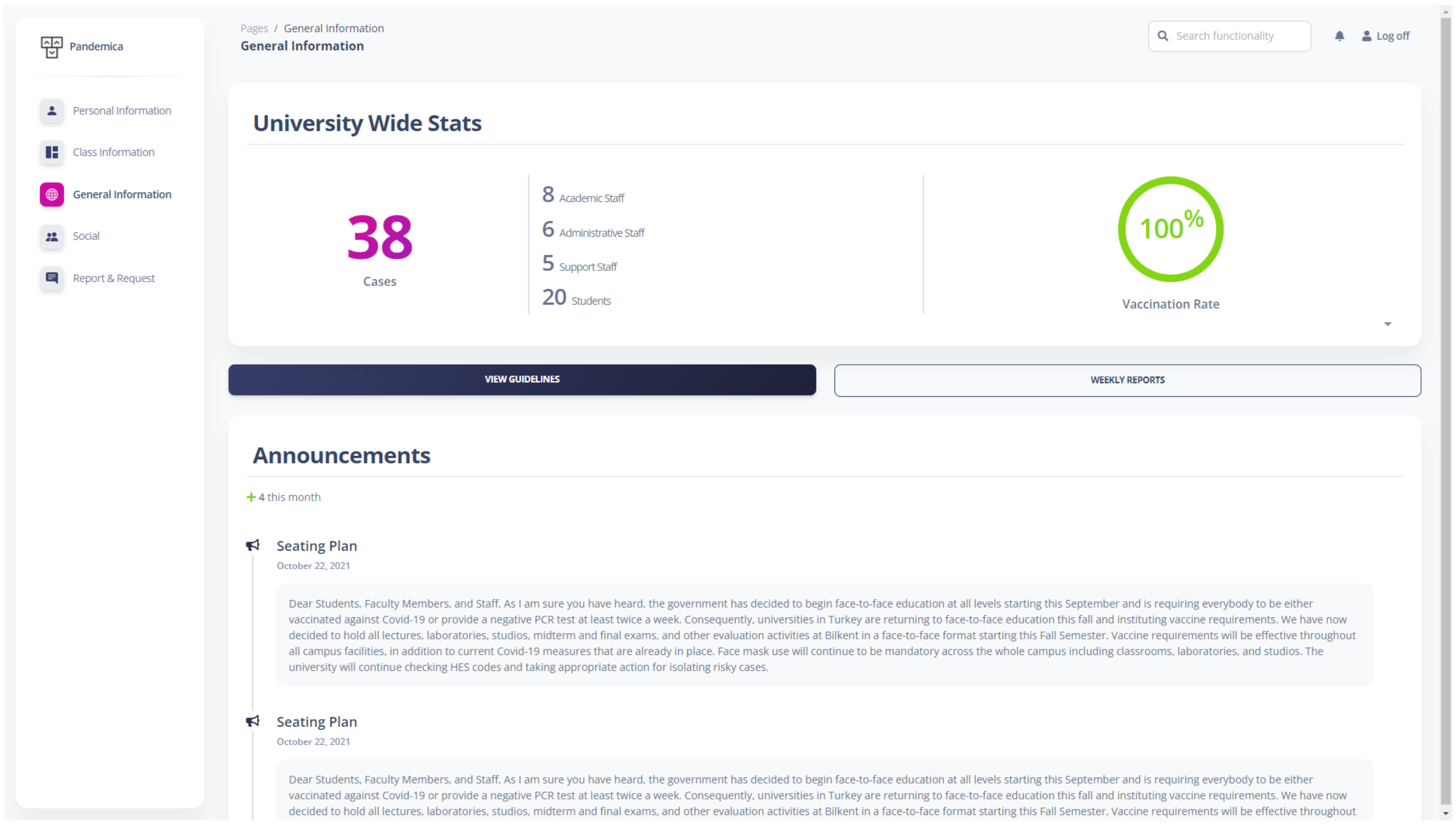


Fig. 16.2: Update Seat Popup

General Information Page

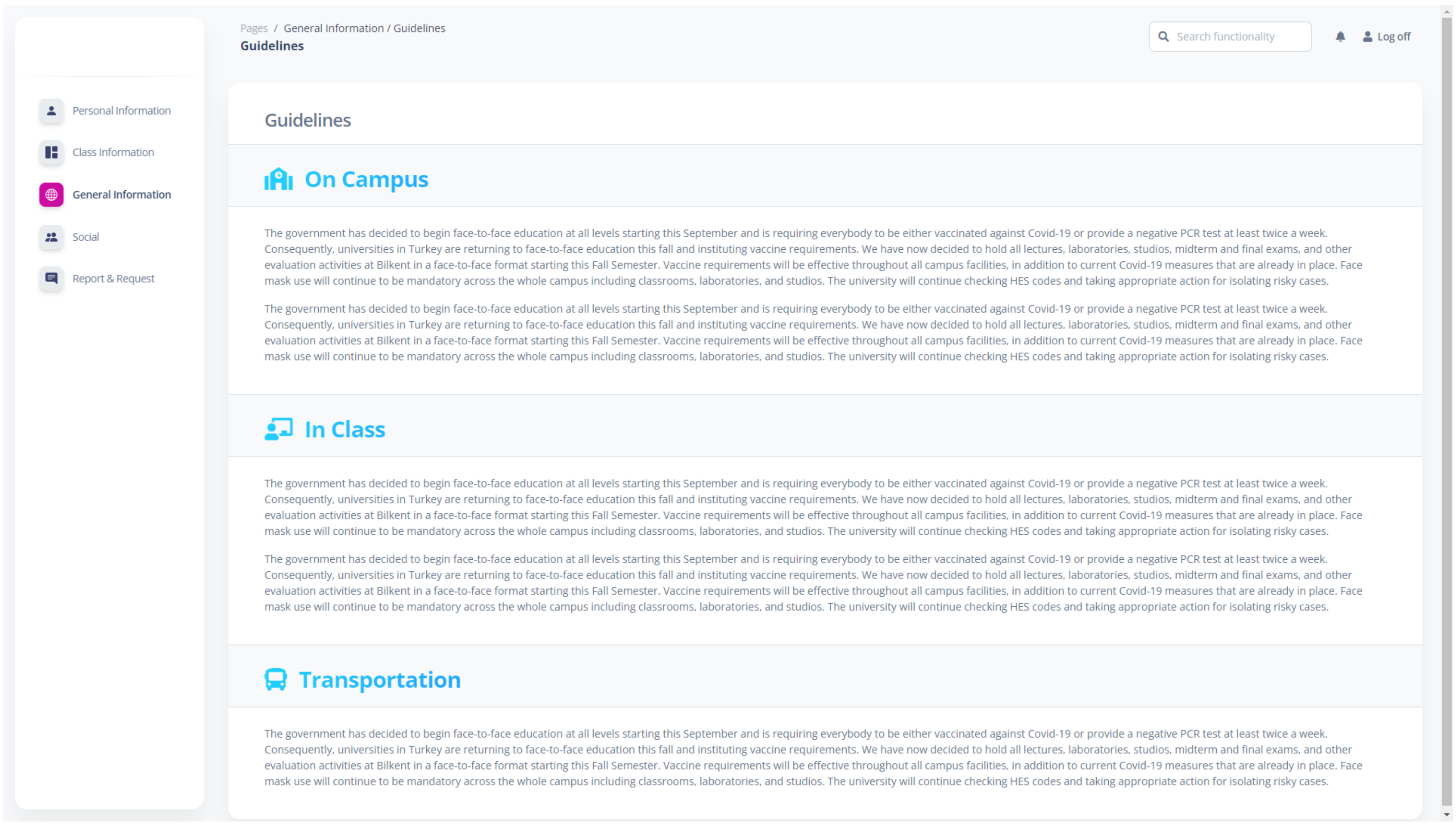


Dear Students, Faculty Members, and Staff, As I am sure you have heard, the government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

Dear Students, Faculty Members, and Staff, As I am sure you have heard, the government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout

Fig. 17.1: General Information Page Mock-up

Guidelines Page


Pages / General Information / Guidelines
Guidelines

Guidelines

On Campus

The government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

The government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

In Class

The government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

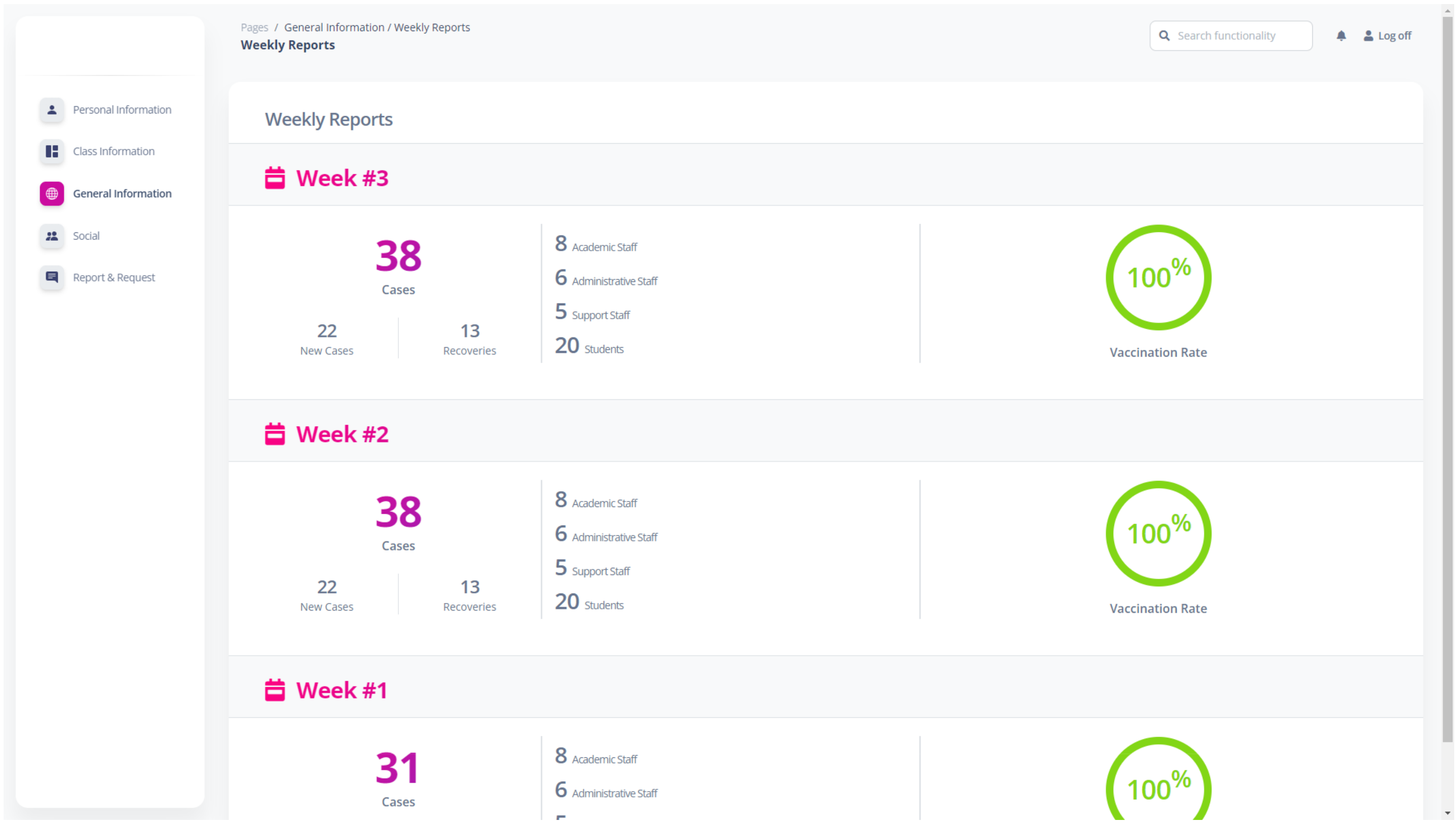
The government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

Transportation

The government has decided to begin face-to-face education at all levels starting this September and is requiring everybody to be either vaccinated against Covid-19 or provide a negative PCR test at least twice a week. Consequently, universities in Turkey are returning to face-to-face education this fall and instituting vaccine requirements. We have now decided to hold all lectures, laboratories, studios, midterm and final exams, and other evaluation activities at Bilkent in a face-to-face format starting this Fall Semester. Vaccine requirements will be effective throughout all campus facilities, in addition to current Covid-19 measures that are already in place. Face mask use will continue to be mandatory across the whole campus including classrooms, laboratories, and studios. The university will continue checking HES codes and taking appropriate action for isolating risky cases.

Fig. 17.2: Guidelines Page Mock-up

Weekly Reports Page



Weekly Reports

Week #3

38 Cases

22 New Cases | 13 Recoveries

8 Academic Staff
6 Administrative Staff
5 Support Staff
20 Students

100% Vaccination Rate

Week #2

38 Cases

22 New Cases | 13 Recoveries

8 Academic Staff
6 Administrative Staff
5 Support Staff
20 Students

100% Vaccination Rate

Week #1

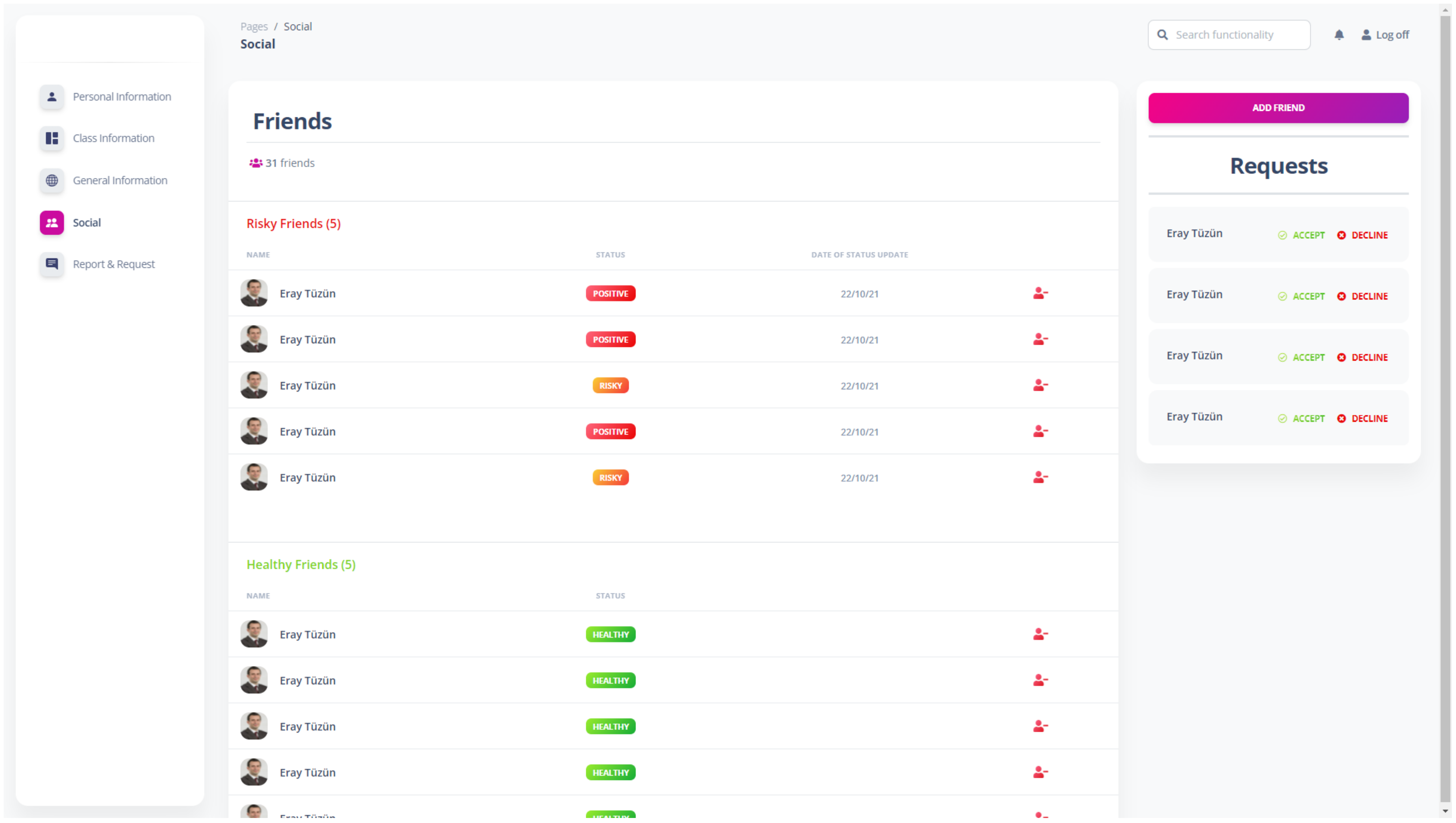
31 Cases

8 Academic Staff
6 Administrative Staff

100%

Fig. 17.3: Weekly Reports Page Mock-up

Social Page



Pages / Social

Social

Search functionality

Log off

Personal Information

Class Information

General Information

Social

Report & Request

Friends

31 friends

Risky Friends (5)

NAME	STATUS	DATE OF STATUS UPDATE
Eray Tüzün	POSITIVE	22/10/21
Eray Tüzün	POSITIVE	22/10/21
Eray Tüzün	RISKY	22/10/21
Eray Tüzün	POSITIVE	22/10/21
Eray Tüzün	RISKY	22/10/21

Healthy Friends (5)

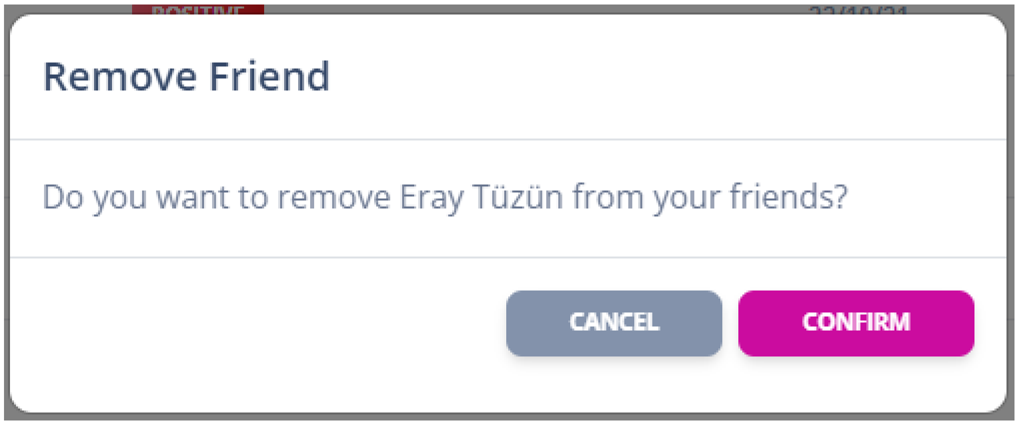
NAME	STATUS
Eray Tüzün	HEALTHY
Eray Tüzün	HEALTHY
Eray Tüzün	HEALTHY
Eray Tüzün	HEALTHY

ADD FRIEND

Requests

Eray Tüzün	ACCEPT	DECLINE
Eray Tüzün	ACCEPT	DECLINE
Eray Tüzün	ACCEPT	DECLINE
Eray Tüzün	ACCEPT	DECLINE

Fig. 18.1: Social Page Mock-up


Remove Friend

Do you want to remove Eray Tüzün from your friends?

Fig. 18.2: Confirm Friend Removal Popup

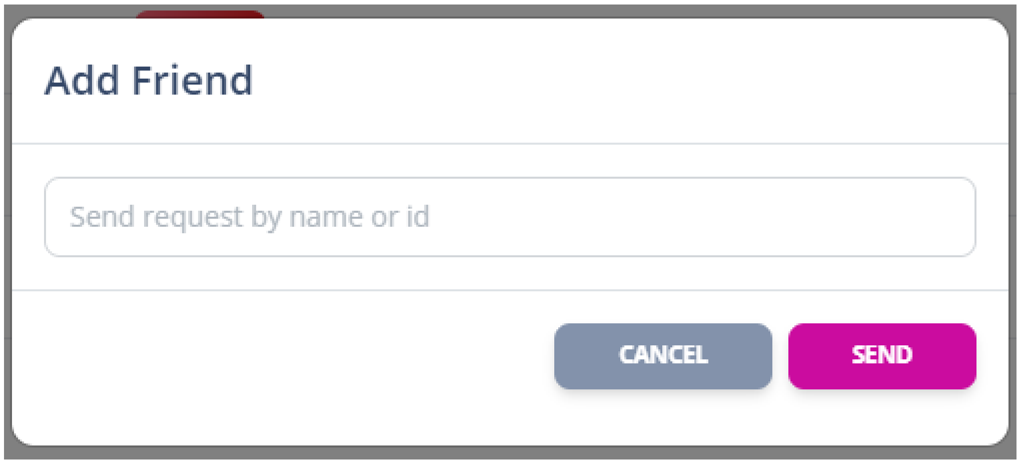

Add Friend

Fig. 18.3: Add Friend Popup

Report & Request Page

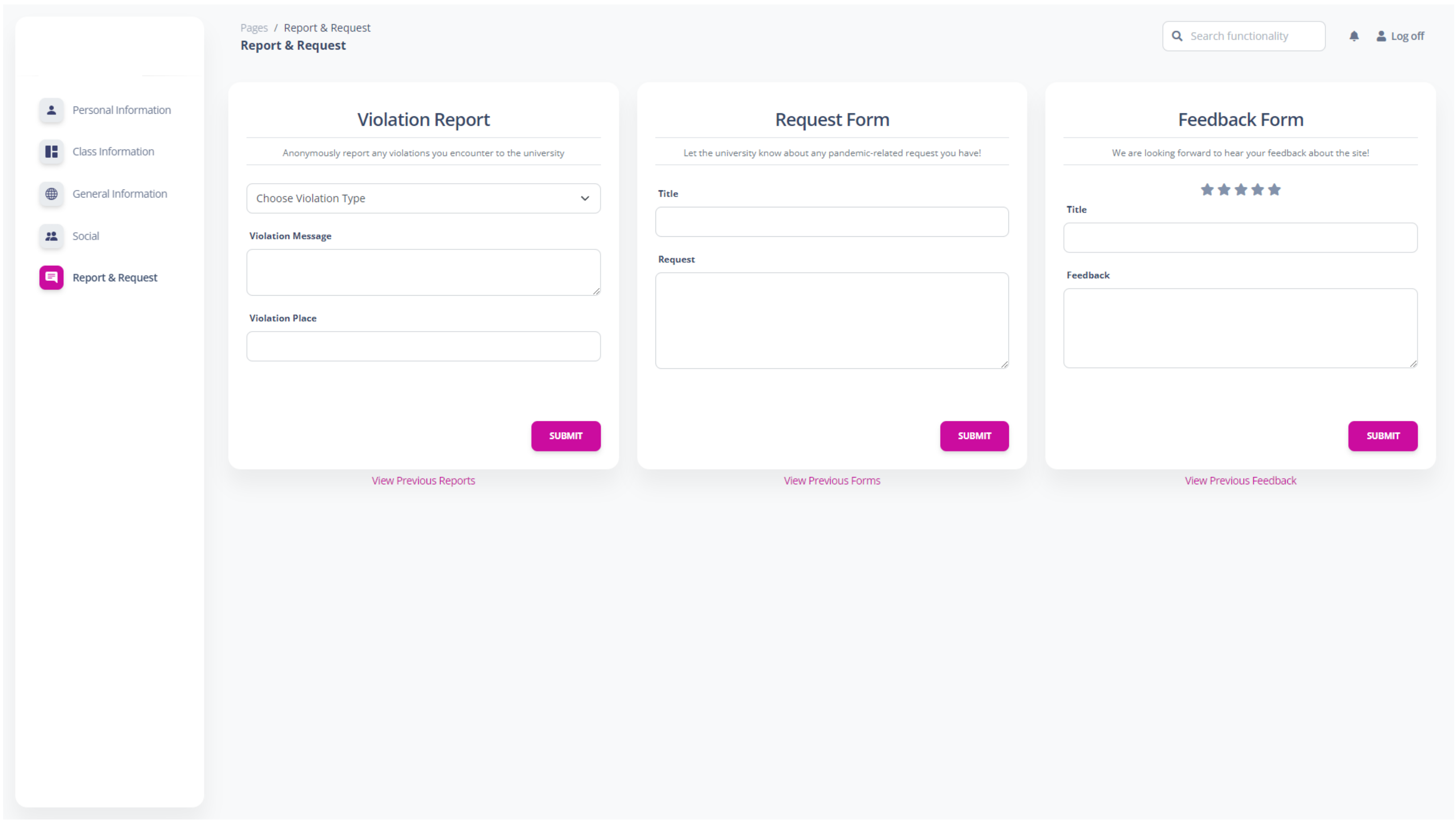


Fig. 19.1: Report & Request Page Mock-up

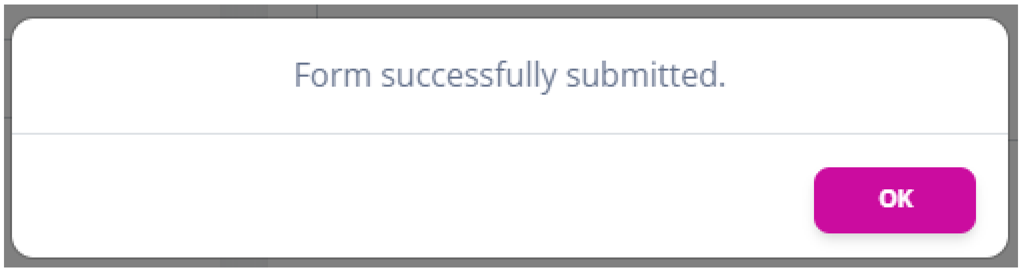

Form successfully submitted.

Fig. 19.2: Form Submitted Popup

Login Page

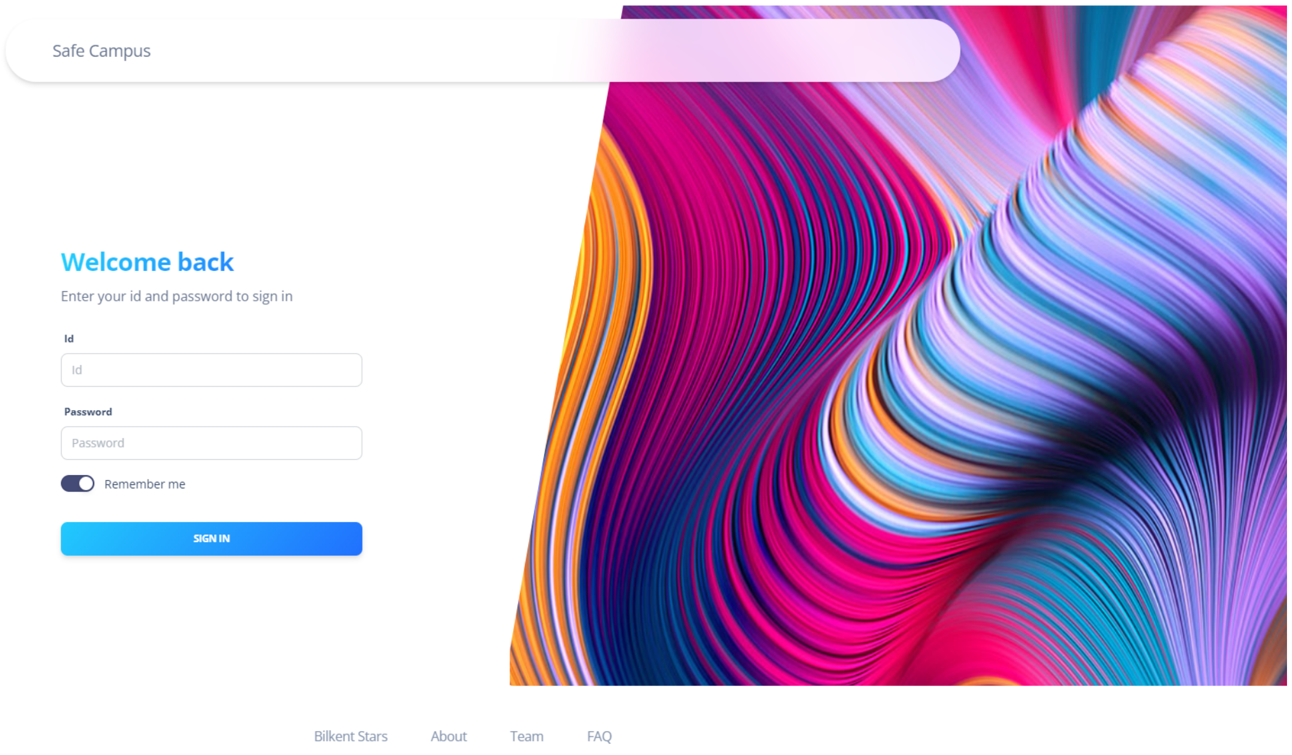
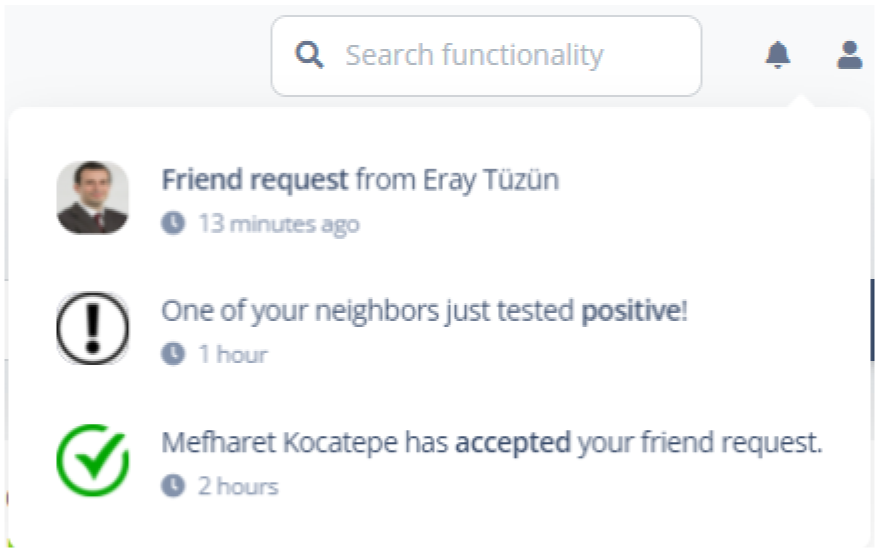


Fig. 20: Login Page Mock-up

Notifications Dropdown



Friend request from Eray Tüzün
13 minutes ago

One of your neighbors just tested positive!
1 hour

Mefharet Kocatepe has accepted your friend request.
2 hours

Fig. 21: Notifications Dropdown Mock-up

References

- Figure 1: https://www.thoughtco.com/contrasting-foreground-background-colors-4061363