Bilkent University

Department of Computer Engineering

CS 319 Term Project

Section 3

Group 3C

Bilkent Erasmus-Exchange System

Analysis Report

Project Group Members

Mehmet Eren Balasar
Mehmet Hasat Serinkan
Tuna Okçu
Yüksel Berkay Erdem
Aftab Fakir

Instructor: Eray Tüzün

Teaching Assistants:
Metehan Saçakçı
Emre Sülün
Muhammad Umair Ahmed
İdil Hanhan
Mert Kara

Contents

1 Introduction 4
2 Current System 4
3 Proposed System 5
3.1 Overview 5
3.2 Functional Requirements 5
3.2.1 University Information 5
3.2.2 Forums 6
3.2.3 Announcements 6
3.2.4 Notifications 7
3.2.5 Direct Message (DM) 7
3.2.6 Fill and Submit Form 7
3.2.7 Search 8
3.2.8 User Profiles 8
3.2.9 User Specific Features 8
3.3 Non-functional Requirements 9
3.3.1 Performance 9
3.3.2 Portability and Compatibility 9
3.3.3 Reliability, Maintainability and Availability 9
3.3.4 Security 9
3.3.5 Usability 10
4 System Models 11
4.1 Use-Case Model 11
4.2 Textual Descriptions of Significant Use Cases 11
4.2.1 Upload File 11
4.2.2 Send DM 12
4.2.3 Upload Student Application Data 12
4.2.4 Fill and Submit Form 12
4.2.5 Approve Courses 13
4.3 Object and Class Model 14
4.4 Dynamic Models 17
4.4.1 Sequence Diagram for Student Nomination 17
4.4.2 Activity Diagrams 18
4.4.2.1 Login Activity 18
4.4.2.2 Student Nomination 18
4.4.2.3 Course Approval Activity 19
4.4.3 State Diagrams 20
4.4.3.1 Login System State Diagram 20
4.4.3.2 Department Coordinator State Diagram 21
4.4.3.3 Student State Diagram 22
4.3.3.4 "TODO" List State Diagram 23
4.3.3.5 Thread State Diagram 24
5 User Interface 25
5.1 Main Page 25

5.2 University Information 26
5.3 Forum 27
5.4 Announcements 28
5.5 Login 29
5.6 Student (applying) Profile 30
5.7 Files (student applying) 31
5.8 Fill and Submit Learning Agreement 32
5.9 Fill and Submit Pre-approval Form 33
5.10 Student (former) Profile 34
5.11 Review Your University (Former student) 35
5.12 Exchange Office Screen 36
5.13 Instructor Profile 37
5.14 Course Approval 38
5.15 Profile View by Other Users 39
5.16 Department Coordinator Profile 40
5.17 Student List 41
5.18 Choosing Students 42
5.19 Exchange Coordinator Profile 43
6 Glossary 43
7 References 43

Analysis Report

Bilkent Erasmus-Exchange System

1 Introduction

As group 3C, we aim to create a web app to complement the already existing https://app.erasmus.bilkent.edu.tr/ website to facilitate the Erasmus application process for both students and administrators (department secretaries, exchange coordinators, instructors, etc.). To this end, we will automate the process of ranking applicants, and make communication between the two parties smoother by providing forums, direct messaging, and form upload capabilities (this way email correspondence will be minimized, as questions can be either sent directly or asked in the forums, and forms are uploaded directly to the website). Some other planned features include user profiles, university information pages, a university review system wherein former Erasmus students can review their universities, automated workload distribution between administrators, an announcements page, and a to-do list for every actor so that each actor is aware of their current responsibilities.

2 Current System

The current Erasmus-Exchange software that Bilkent University uses only serves as an application and information system. Furthermore, except the application system (app.erasmus.bilkent.edu.tr), there is not a comprehensive, all-in-one website that could help all students and administrative staff manage the long Erasmus-Exchange processes. Every department has its own Erasmus information page along with Bilkent's generic Erasmus-Exchange pages scattered around the internet. However, the most critical problem is that there are no systems to support the stakeholders during the Erasmus-Exchange periods. In other words, users utilize information software and application software, then, during and after the mobility period all operations are done manually without the help of an Erasmus-Exchange system. We will try to combine existing systems and implement new ones to make everyone's life easier.

3 Proposed System

3.1 Overview

Bilkent Erasmus-Exchange System is an all-in-one web-based application that will help students, coordinators and all other users who are somehow involved in the Erasmus-Exchange procedure in Bilkent have a smooth and spotless Erasmus-Exchange experience. The main input data sources of this system will possibly be Bilkent SRS (Student Registration System), Bilkent Exchange Office and users directly. SRS will be used to authorize and verify users and Exchange Office will upload the application documents generated by "app.erasmus.bilkent.edu.tr". There will be different tabs for different functionalities:

3.2 Functional Requirements

3.2.1 University Information

This page is designed specifically for students or other people who are curious about the universities that Bilkent offer for an Erasmus-Exchange program. It will be specific to each department since every department has their own agreements with different universities. Along with the general information -location, brief history, offered courses etc.- about universities there will be more advanced functionalities listed below:

- Rating of a university: Rating is calculated as an average of the ratings given by former students that have gone this institution and came back. Rating points will be out of 5. Of course, it will not be valid as an official rating given by an Erasmus-Exchange authority for a university, but it will give a rough indication as to how is that universities Erasmus-Exchange program.

- Estimated point thresholds: For each university, by using the data from previous years, system will calculate a point average and estimate an acceptance threshold for that university. For example, let's say, in the previous years, accepted students had 74, 73, 72, 71 points. The average is 72.5 (This is just a simple algorithm that will be replaced with a more advanced one). Threshold will be shown as 72.5 and students who want to select this institution will have an idea. It should be noted that these point thresholds

are only estimates and they are on the page only to give a rough idea about the university. We observed that a lot of students applying for the programs do not know how many points they need to get accepted, or sometimes they waste their hard-earned points on lower-level universities since often it is hard to learn if a university is of high quality. However, there will be very obvious warning signs about the fact that these are just estimates on the page.

- Erasmus-Exchange point calculator: Users will be able to manually input their related grades and CGPA to calculate their Erasmus-Exchange points to see whether they can be accepted to a university. Again, all these processes are based on just estimations and give a rough indication.

- Reviews: Former students will be able to comment on their experience during the Erasmus-Exchange program. They are encouraged to enter reviews about their universities and rate them. These reviews will be shown beside the universities.

3.2.2 Forums

One of the main problems of the current system is that every year new students apply for Erasmus and same questions are asked to coordinators. There are Q&A pages, but they are not enough, and they are not interactive. Students will be able to ask questions and start threads on forums specific to departments. They will be able to view all forums, threads, and answers though. Only users with management roles and former students will have the permission to answer. Most viewed questions, most recent questions will be shown to the user to make them find help easier and prevent them from asking the same questions, which may put too much stress on the database.

3.2.3 Announcements

Management users will be able to make announcements and inform all other users about a generic situation. This feature is needed mainly because currently everyone is informed via e-mail, so it is messy and really difficult to keep track of the information chain. Possibly, an e-mail will still be sent automatically to those the

announcer selects. Guest users (for extensive information about actors, see 5.3) will also be able to view announcements.

3.2.4 Notifications

A notifications panel that will inform the user about every change somehow related to them will also be implemented. For each user type, there will be differences as to which notifications they will get. Announcements, answer to a question asked, new files uploaded, new messages etc.

3.2.5 Direct Message (DM)

Perhaps the most important part of our system is the Direct Message feature (will be referred as DM from now on). Correspondence between the actors of Erasmus-Exchange system is another one of the main problems because there is not a domain-specific communication tool. In our DM system, every logged-in user can send text messages to another logged-in user. DM will be available from the moment the user logs in and from every page of the website. Unfortunately, we are not planning to implement a feature that enables sending messages in other formats such as “jpeg”.

3.2.6 Fill and Submit Form

Most of the documents will be uploaded as “pdf” files, so data inside these files will not be extracted and processed in the database. However, the most important documents, which are Preapproval Form and Learning Agreement, will be generated by our website rather than being uploaded by a user. Input fields of these forms shown on the related page will be filled by the user and the rest will be handled within the system.

There are specific reasons why these two documents are treated differently than others. As mentioned above, data (specifically course data) from Preapproval Form is needed inside the system for various other functionalities (for example, instructors will have a page to approve all host institution courses proposed by students etc. – look at the mock-up for more info on 5.5.9). Learning Agreement is also generated by our

system not because we need the data but because the formatting of this document is complicated so it would be easier for users to just fill data fields.

3.2.7 Search

The search functionality will guide users inside the website. It can be mainly used to search for student names, coordinator names, university names, countries, files, services, pages, and questions. We are planning to make it cover almost all data on the database (not private data).

3.2.8 User Profiles

Every actor will get a different user profile according to the functionalities they can use. All information about the user including every file they have uploaded will be on the profile page. Also, users will be able to see their "TO-DO" lists on their own profile pages (a user's "TO-DO" list is not shown to another user). For detailed information about a specific actor's profile page, you can look at 5.5.6.

3.2.9 User Specific Features

There are a lot of actors who will be using the system. Of course, there are some functionalities that only certain actors can use:

- Course approval: Every instructor is responsible for deciding whether a course from the host university can be exempted and the course that instructor teaches corresponds to this course. For this purpose, we will implement a page showing all the proposed courses from various students going to various institutions that possibly correspond to an instructor's course. The instructor will review these courses and give exemptions.
- Student list: A department coordinator will be able to see all the students he/she is responsible for in one page and see what they have uploaded or not uploaded yet (look at 5.5.17 for visualization).

3.3 Non-functional Requirements

3.3.1 Performance

The performance of the website is one of the most important issues. We will aim for the website to be loaded in 2 seconds when the simultaneous user count is less than 5.000. If the simultaneous user count exceeds 5.000, the time for the loading website will be closer to 4 seconds as the user count closes to the 20.000 limit. To accomplish that, the application must have small-sized image extensions like ".webp" and a well-organized database.

3.3.2 Portability and Compatibility

The application must be able to be implemented on different browsers at any time. When the users use the application through A browser and change the browser to B, the application must behave in the same way on browser B. Also, It is considered that the application must use new libraries and frameworks which make the application useful and easily understandable. So, the versions of browsers that our application can be implemented on are listed below:

- Chrome (version 49) (release: 2016)
- Firefox (version 50) (release: 2016)
- Safari (version 10) (release: 2016)
- Internet Explorer (version 10) (release: 2012)
- Edge (version 14) (release: 2016)

3.3.3 Reliability, Maintainability and Availability

Our application will be performed with success in 95 percent of the general use cases and success in 99 percent of the most important use cases such as approving or rejecting cases. If the application is in a failure, the time to fix it must not be greater than 1 hour. The users can access our web application at any time every day.

3.3.4 Security

The information stored in our database must be encrypted to be sure about their security. Every encryption and decryption algorithm will be generated randomly to be

specific for that user. There must be one base secret key for implementing encryption and decryption, but every user's secret key must be randomly created from this key. Because every user has a different secret key, even if one user's encryption method is exposed, other users' data must remain safe. The ability to access the database must be authorized to people who have an "Admin" role. In addition to this, the application must store log records on the database.

3.3.5 Usability

Our application must be useful to new users. Therefore, the designs of the pages must be easy to understand and use. The “Exchange” application must meet this need by designing the pages according to new generation methods such as “Tab Bar Menu” and “Pop-up Menu”. In this way, the users can see whatever they need without any detailed search. Also, the application must have different interfaces for different authority levels to go away any kind of complication. Most used functions of the application must be located in the main perspective of the user’s view on the pages.

4 System Models

4.1 Use-Case Model

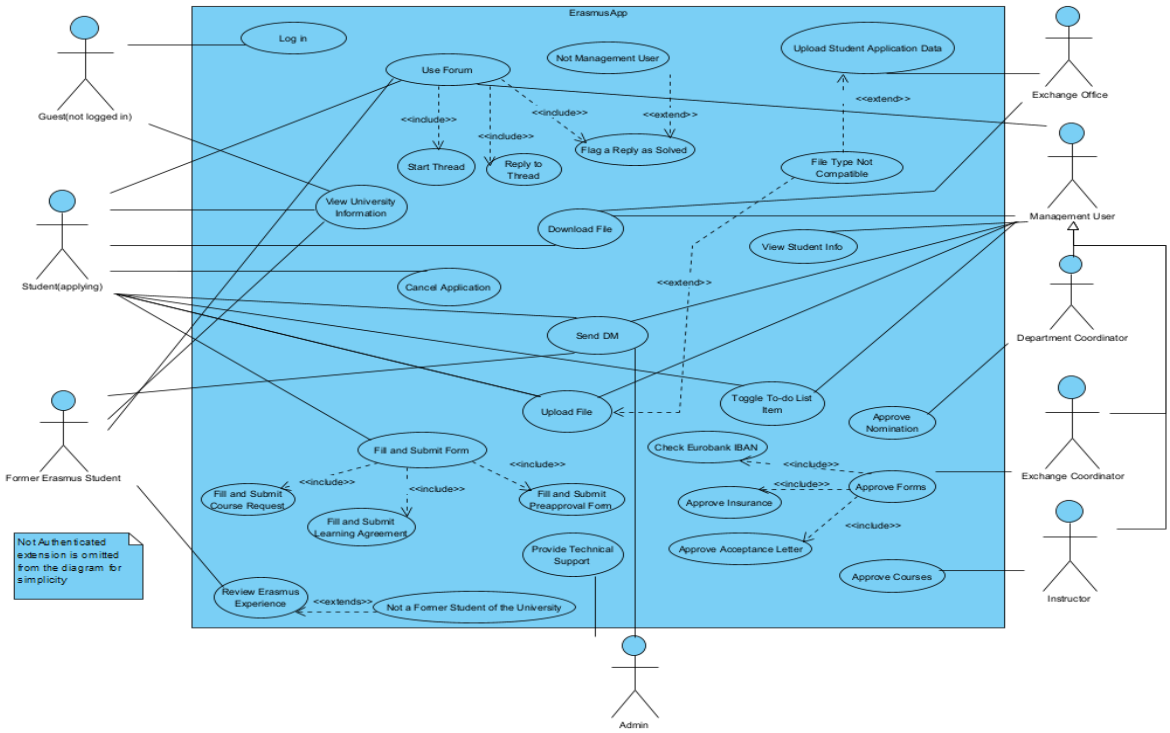

Figure 4.1.1 Use Case Diagram

4.2 Textual Descriptions of Significant Use Cases

4.2.1 Upload File

Name: Upload File

Participating Actor: Student, Management User

Entry Condition: User is logged in

Exit Condition: None

Flow of Events:

1. Students click the upload button
2. Students name the artifact, determine the artifact's type, determine the artifact's type.

3. Students choose the file to be uploaded.
4. Students click the submit button to upload the artifact.
5. The system returns an appropriate error or success message.

4.2.2 Send DM

Name: Send DM

Participating Actor: Student, Management User

Entry Condition: User is logged in

Exit Condition: None

Flow of Events:

1. User clicks the send DM button.
2. User types the message into the text field.
3. User click send button to send message.

4.2.3 Upload Student Application Data

Name: Upload Student Application Data

Participating Actor: Exchange Office

Entry Condition: User is Exchange Office

Exit Condition: None

Flow of Events:

1. User clicks the upload button.
2. User selects the file.
3. User click the submit button to upload file.
4. The system returns an appropriate error or success message.

4.2.4 Fill and Submit Form

Name: Fill and Submit Form

Participating Actor: Student,

Entry Condition: User is Student

Exit Condition: None

Flow of Events:

1. User clicks the fill and submit Learning Agreement button or fill and submit Pre-Approval form button.
2. User types the necessary information into input fields.
3. User either submit it or save it.
4. The system returns an appropriate error or success message.

4.2.5 Approve Courses

Name: Approve Courses

Participating Actor: Instructor

Entry Condition: User is Instructor

Exit Condition: None

Flow of Events:

1. User clicks the approve courses button.
2. User select the student.
3. User either approve the course or deny it.
4. The system returns an appropriate error or success message.

4.3 Object and Class Model

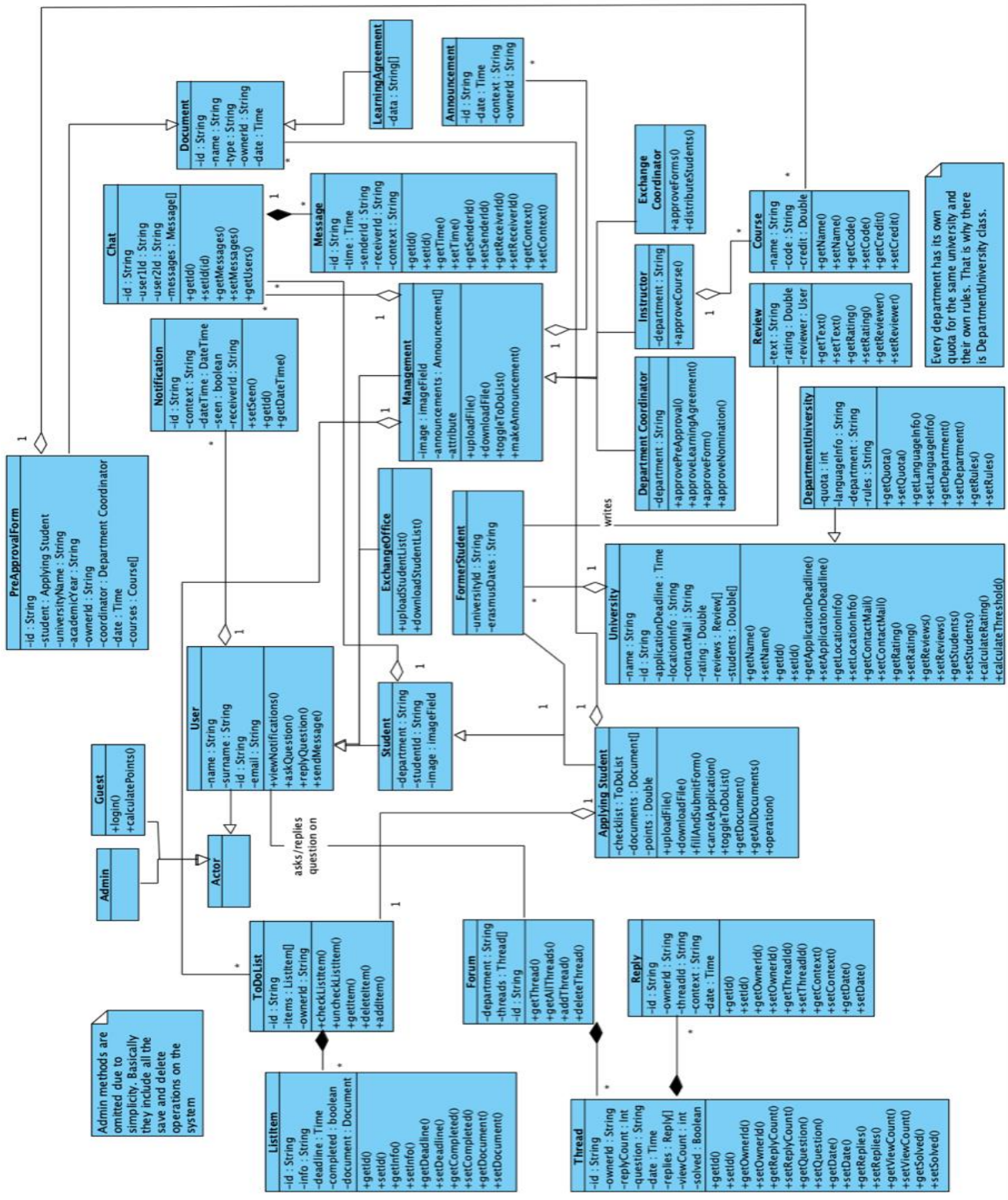

Figure 4.3.1 Class Diagram

The class diagram of Bilkent Erasmus-Exchange System is shown above. The class diagram has 30 classes currently. These classes, their attributes and their functions are created considering users' needs and authority levels. Purposes of some classes are obvious, so they are not included in this explanation.

First of all, we have two different main classes which are "Actor" and "System". Actor class is created to organize the authority hierarchy in the application process and maintain the "Exchange" app. Also, this class has 3 subclasses called "Admin", "User" and "Guest" whose details are explained below.

As mentioned above, there is also "System" class which enables the user to use our application's functions properly. We decide to divide the whole code into 2 different main classes because the processes of implementation and maintenance will be easier before and after release.

Now, we start to talk about subclasses of the Actor class:

Admin:

Admin class is created to develop and maintain the Exchange application. This class has access to change the application's code and get/post data from/to the database with some limitations which secure the privacy of the user. There are no attributes and functions of this class since the Admin class is not an actual stakeholder of the project.

Guest:

This class enables the user to view a couple of pages without login (e.g Forum page). They cannot upload or download anything. Also, they cannot see the pages which are exclusive for the logged-in users. This class is a representation for out of campus students and not applied students yet. When they log in, they transform into the User class. The team observed that a lot of people are just curious about Erasmus and Exchange programs so we decided that some of the features, mainly information pages, should be available to them without bearing the burden of logging in.

User:

Basically, this class is one of the very important classes of our diagram. Whether the user is an authority or not, he/she is a part of this class. The User class has some

important attributes such as name and email. Also, this class has 3 subclasses which are Exchange Office, Management and Student.

Exchange Office:

This class is a very simple class. Exchange Office can only upload the students' data and download files about the application process.

Management:

Management class represents the coordinators of Erasmus programs in Bilkent except the Exchange Office. Members of this class can follow the application, nomination, mobility and return processes and use special functions of our application to organize and manage the process. Such a parent class is needed since most of the functionality is shared between all three subclasses of Management but still different UI's and specific functions are required.

Student:

This is the class that represents all applied students or students who completed their Erasmus/Exchange process and came back. The Student class is designed to make the application process easy for both the applying students and management people. This class have 2 subclasses:

Applying Student:

Applying Student class has two different and important attributes which are called checklist and documents in the diagram. Thanks to these attributions, the applying students will know what he/she must do, and the authority people can see which documents belong to that student. Also, this class has functions about the application process so that the user can easily implement their application.

Former Student:

This class is a representation of the ones who completed their Exchange Program and the difference is that they can review the Erasmus experience while there is no class that can do it.

We have also the “System” class which is one of the main classes as mentioned above. In the System class, we have three main subclasses which are called “Contact”, “Info” and “Procedure”. The main reason for this is that we can easily add or delete different attributes or functions to classes according to their specific area.

University:

The stuPoints attribute is a data structure to store points of former students who are accepted to this university. These points are stored because they will be used to calculate an estimated acceptance threshold for this particular university.

4.4 Dynamic Models

4.4.1 Sequence Diagram for Student Nomination

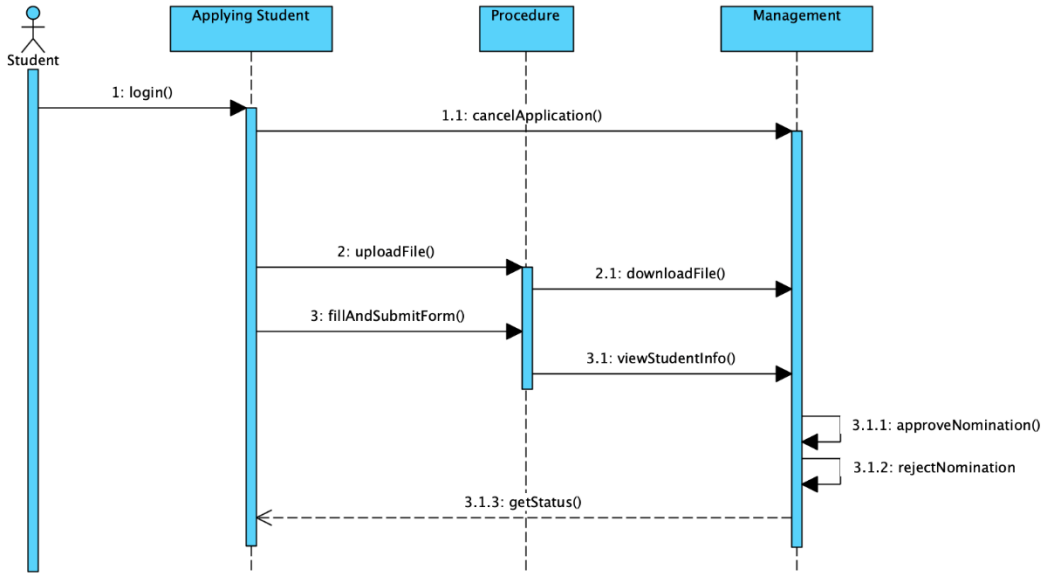

Figure 4.4.1.1 Sequence Diagram

4.4.2 Activity Diagrams

4.4.2.1 Login Activity

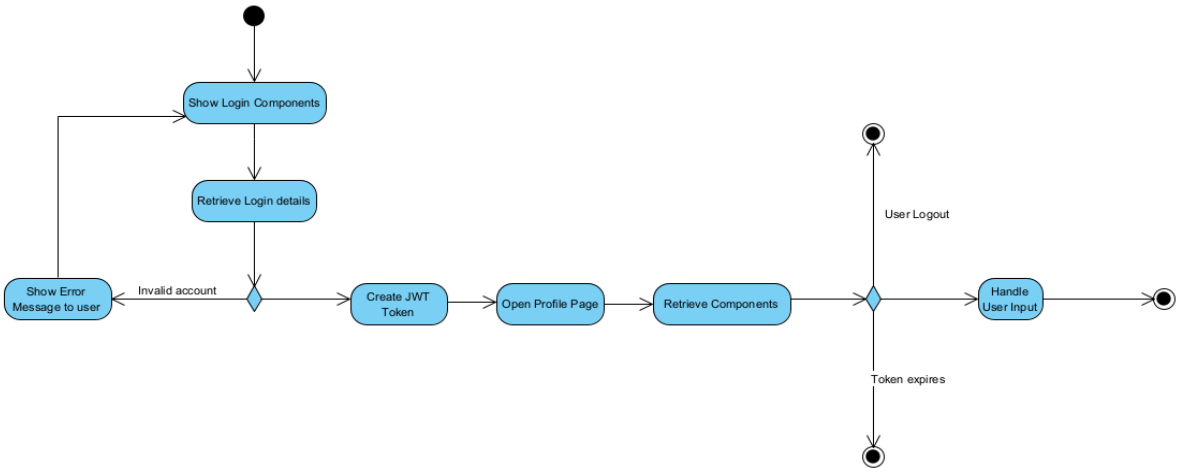

Figure 4.4.2.1.1 Login Activity Diagram

4.4.2.2 Student Nomination

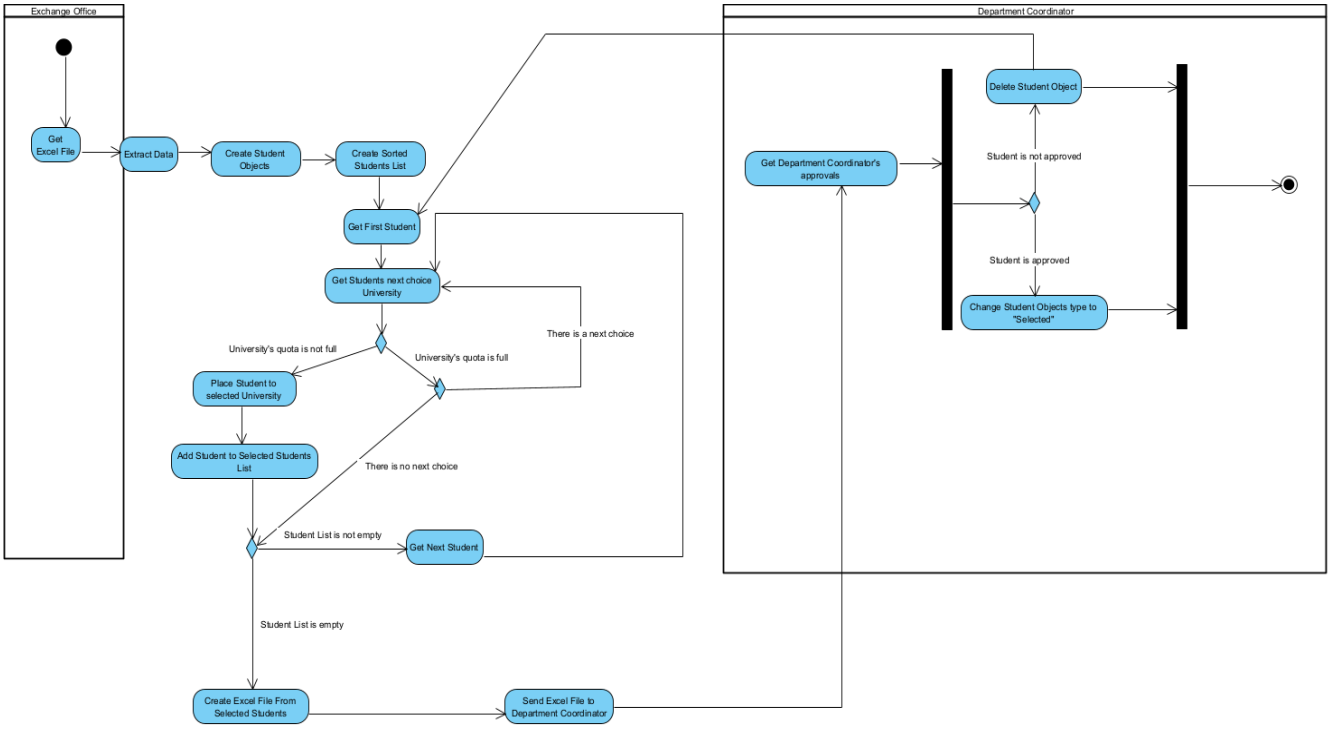

Figure 4.4.2.2.1 Student Nomination Diagram

4.4.2.3 Course Approval Activity

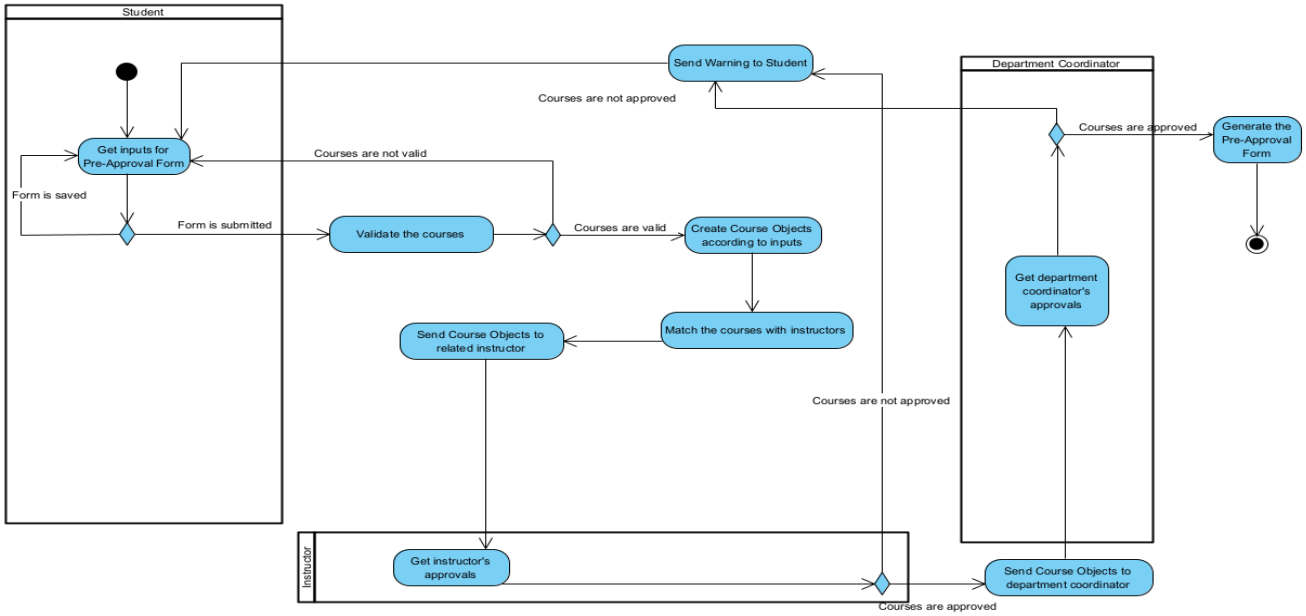

Figure 4.4.2.3.1 Course Approval Diagram

4.4.3 State Diagrams

4.4.3.1 Login System State Diagram

The viewer can attempt to log in, in which case the system checks their information against the information of the users in the database. If it doesn't match any user, their viewer status is unchanged. Else, depending on which user the information inputted matches, the viewer is matched with a role and is provided appropriate privileges. (Note that this matching process is unrepresented in the diagram for brevity.) Then, the user can log off whenever they'd like.

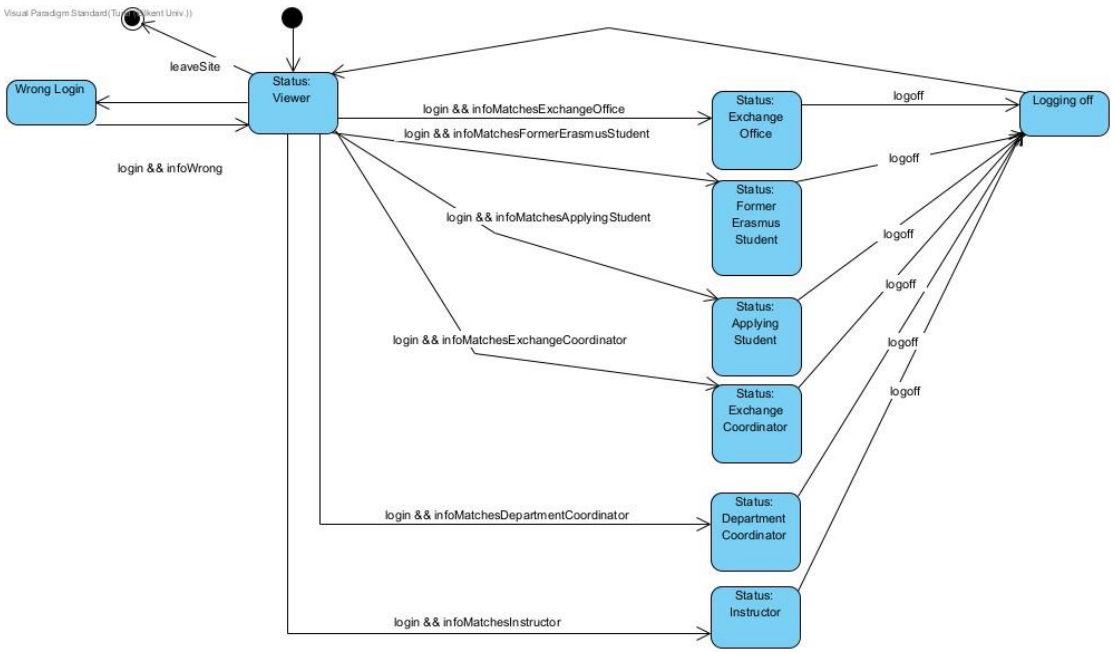

Figure 4.4.3.1.1 Login System State Diagram

4.4.3.2 Department Coordinator State Diagram

Once the student data is finalized by the exchange office, the students are ranked by the system and those to be nominated are promptly chosen. Once this happens, the department coordinator becomes active and can either approve or reject nominations or cancel a nomination choice so that it may be remade. When the nomination list is exhausted (i.e. every student nominated is either approved or rejected), nominations can be finalized, at which point the state machine becomes inactive again.

Note that as in any other state diagram representing a role, the basic capabilities of this role are not included in this diagram, as they do not advance the state machine to a new state. For example, a department coordinator can send DMs and use the forums, but as these actions do not change its state, they are not represented here. Note also that these actions can be used even in the inactive state.

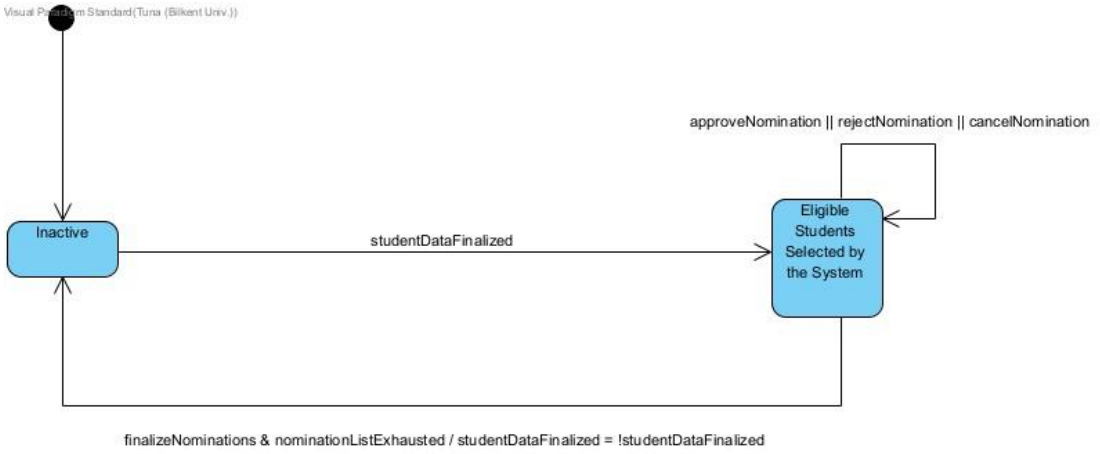

Figure 4.4.3.2.1 Department Coordinator State Diagram

4.4.3.3 Student State Diagram

If a student is already in the system as a former Erasmus student, they stay as one. If the excel file uploaded includes this student, they are now in the system as an applicant (Did Apply). Everyone else is a non-applicant. If a student goes through the entire Erasmus application process successfully, they become a former Erasmus student and stay as one.

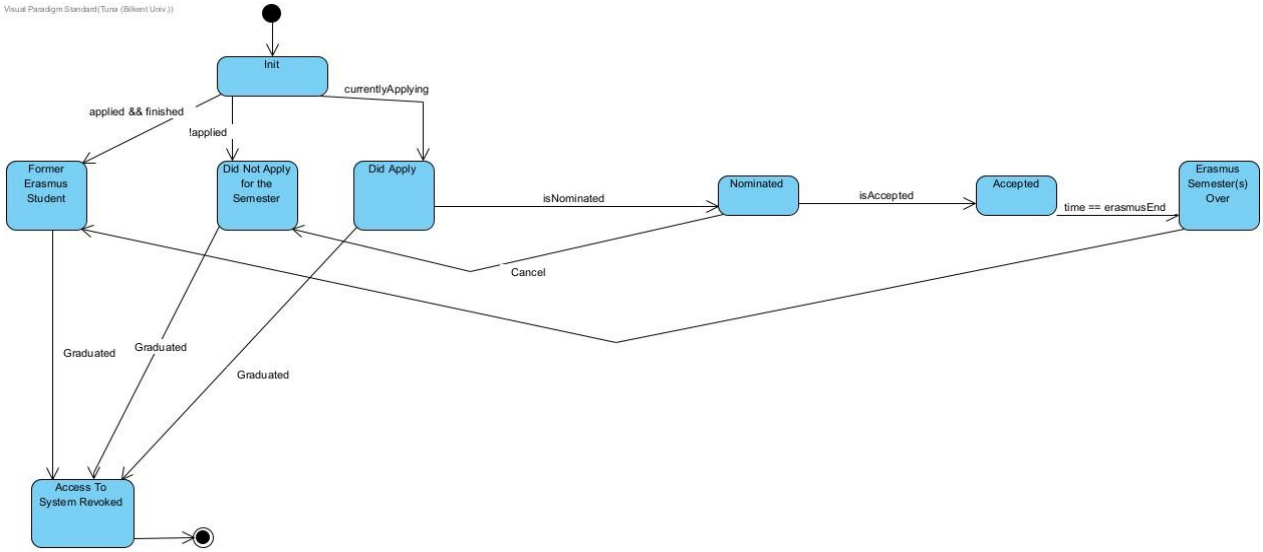

Figure 4.4.3.3.1 Student State Diagram

4.3.3.4 "TODO" List State Diagram

Each to-do list item included under a to-do list is generated by the system: there is no way for a user to create one of their own. Each item starts out unchecked and is automatically checked whenever the system detects that the responsibility the item corresponds to has been fulfilled (ex. course preapproval form uploaded). Then, if the user feels like they should redo this part, they can uncheck the item themselves after which they will be able to check it back themselves if the responsibility is fulfilled. (So initially the item is unchecked, which is automatically checked when the course pre-approval form is uploaded. If the user realizes they made a mistake, they can then uncheck the item themselves, after which they can immediately check it back. However, if they also retract the form after unchecking the to-do list item, the system won't allow them to check the item again so as to prevent a mistake. When they upload it again, they will be able to check the list themselves.) Once the deadline for the responsibility is reached, if the item has been checked already, the item terminates and is removed from the list. If not, the item transitions into the "Task Not Completed Within Deadline" state, signaling the high priority of the task (the user is allowed the option for late submissions). Once this responsibility is fulfilled, the item terminates.

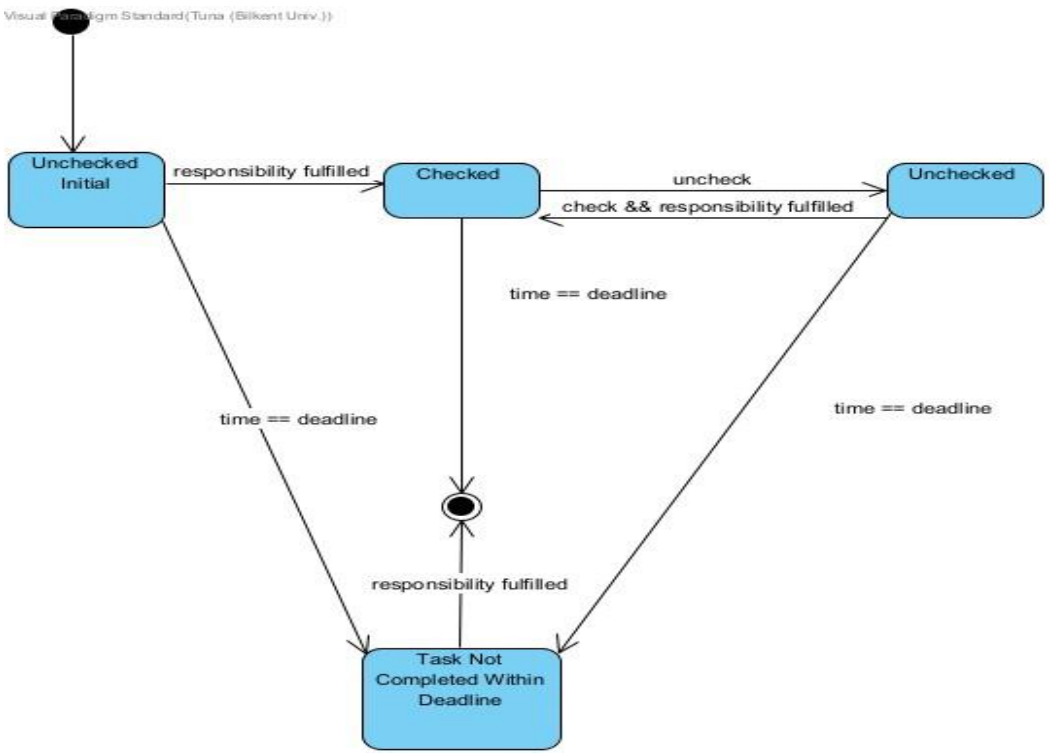

Figure 4.3.3.4.1 Todo List State Diagram

4.3.3.5 Thread State Diagram

A user may open a forum thread to ask a question under which other users can attempt to answer the said question. An admin can(in addition to replying) discard the thread or mark it as solved(while higlighting the most useful answer), closing down replies. A solved thread will still be visible on the forums. However, since, an admin might notice something wrong with a solved thread, solved threads can also be discarded if needed.

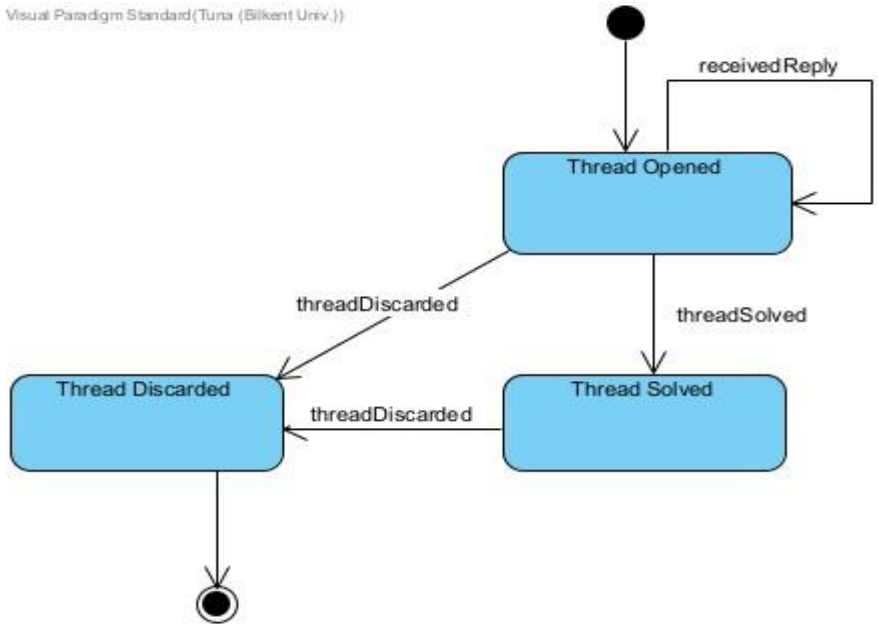

Figure 4.3.3.5.1 Thread State Diagram

5 User Interface

5.1 Main Page

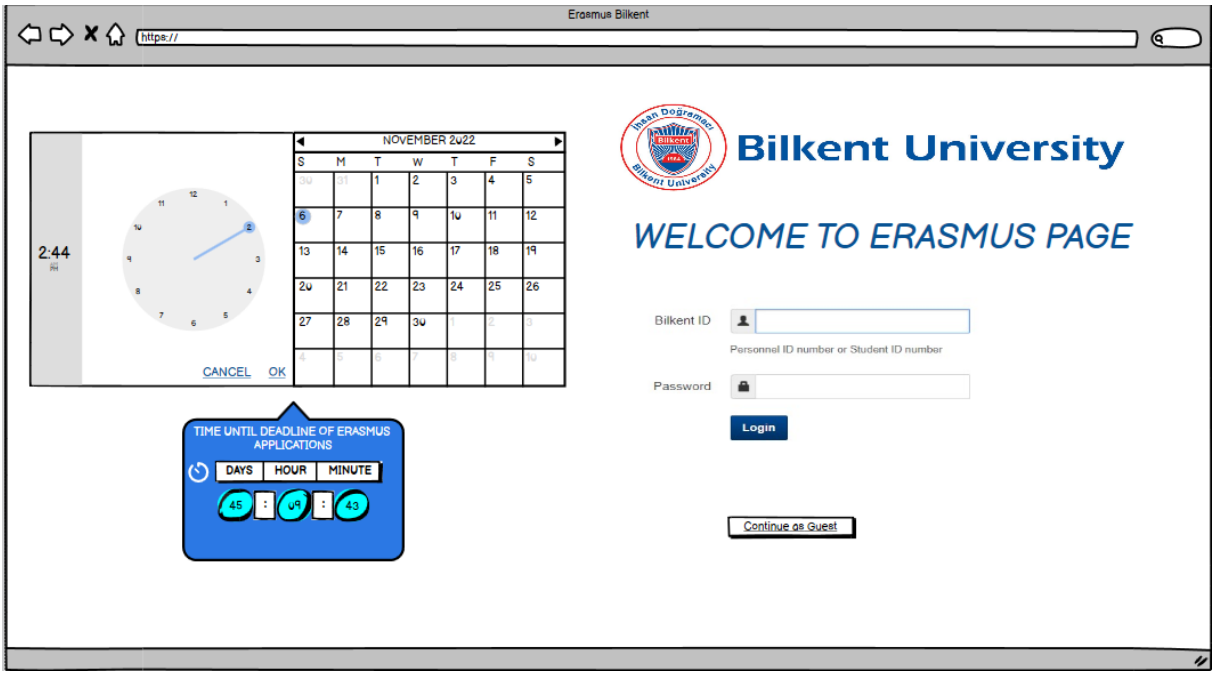

Figure 5.1.1 Main Page Mock-up

The main page will be shown first when the user enters the website. The time until the deadline for ERASMUS applications will be shown on the left part of the main page. On the right side, since users can log in through SRS, the SRS login will be there. Also, users can use the website without logging in by pressing Continue as Guest button.

5.2 University Information

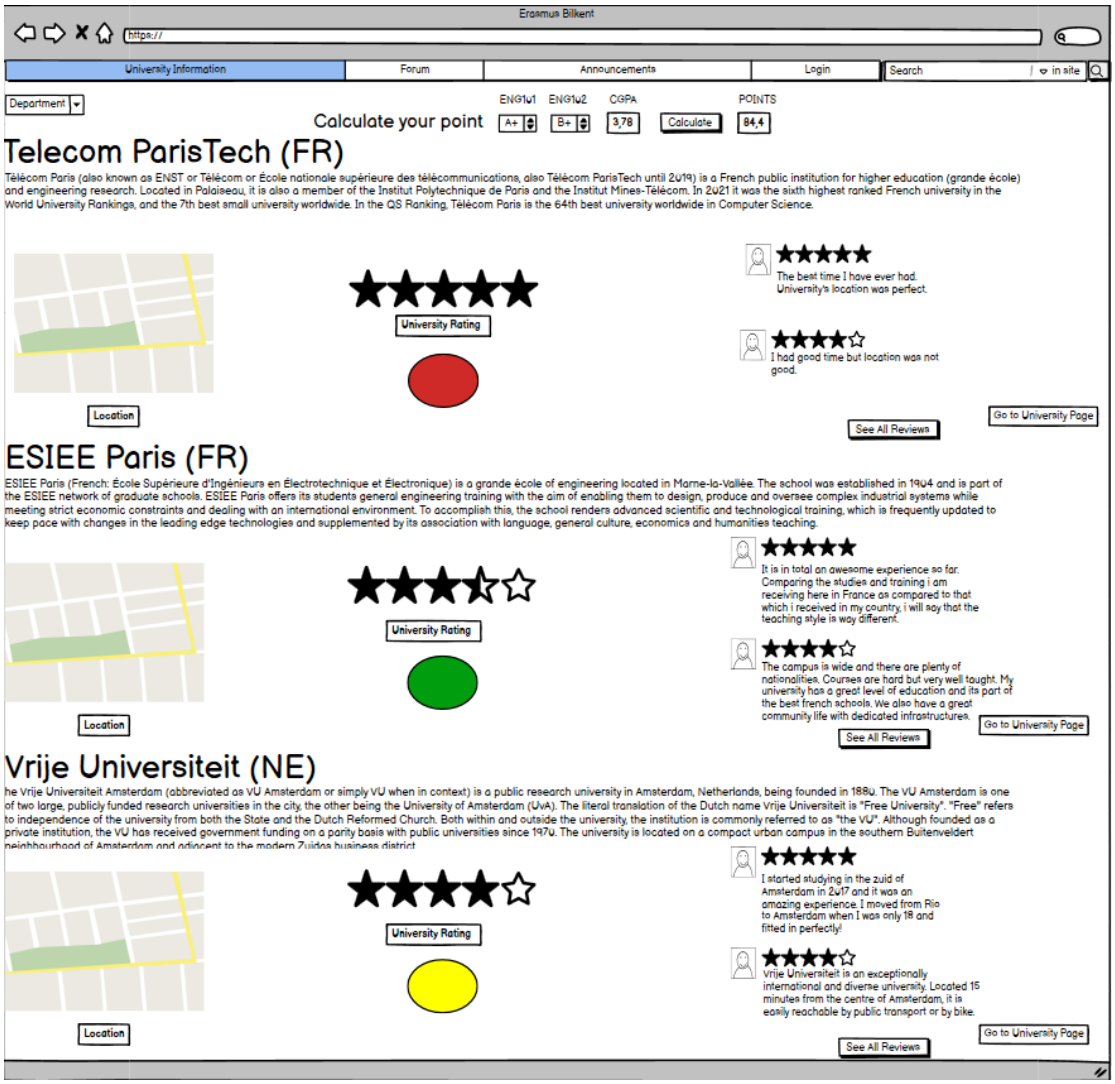

Figure 5.2.1 UniInfo Mock-up

When the user continues as a guest, the university information page will be opened. On this page, users can see all the possible ERASMUS universities of the particular departments. Users can see reviews, the points of former students, location, university ratings and users can go to university's webpage by button. Moreover, at the top of the website, there will be an ERASMUS point calculation area. When the calculation is done, the circles that are shown in each university information will be:

- Red: It means that the user is not likely to be selected for that university.
- Yellow: The user may enter that university.
- Green: The user might be easily selected for that university.

5.3 Forum

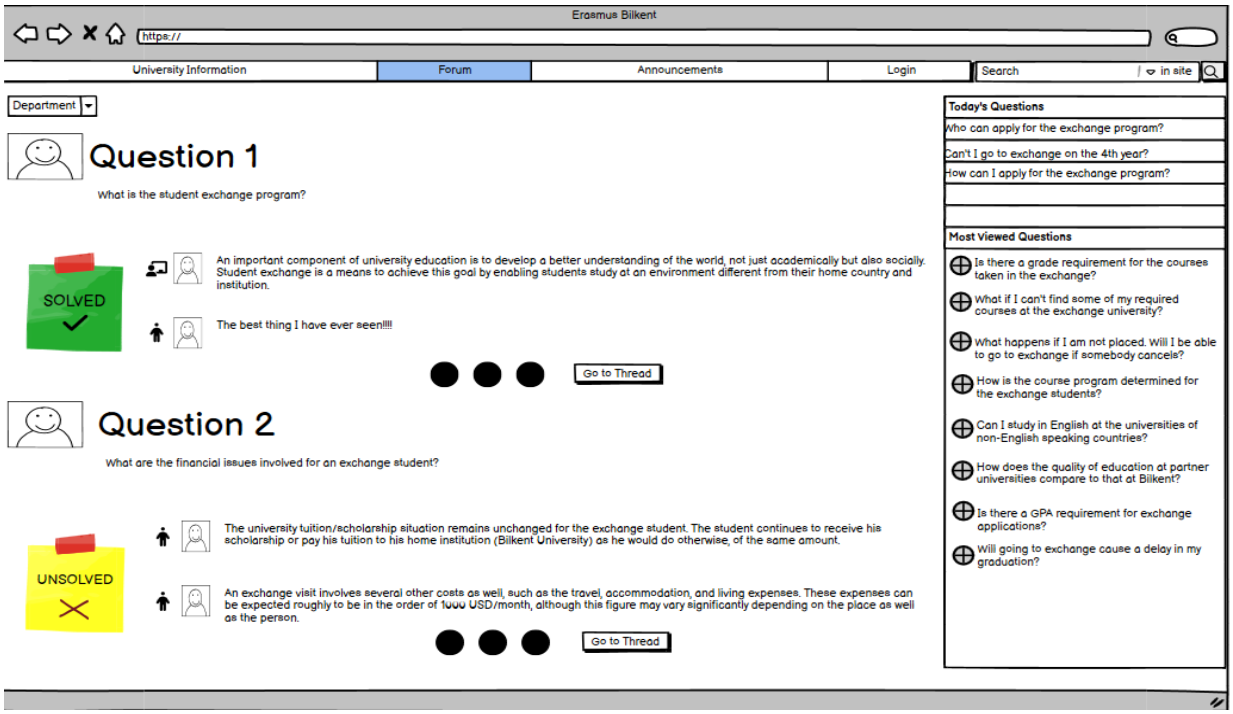

Figure 5.3.1 Forum Mock-up

In the forum, users can see the questions. They are separated by departments. On each question, whether it is solved or not will be shown. Users can see the thread of the question by using Go to Thread button. On the right part of the forum, the questions asked today and that are most viewed will be shown. Users can easily jump to that question's thread.

5.4 Announcements

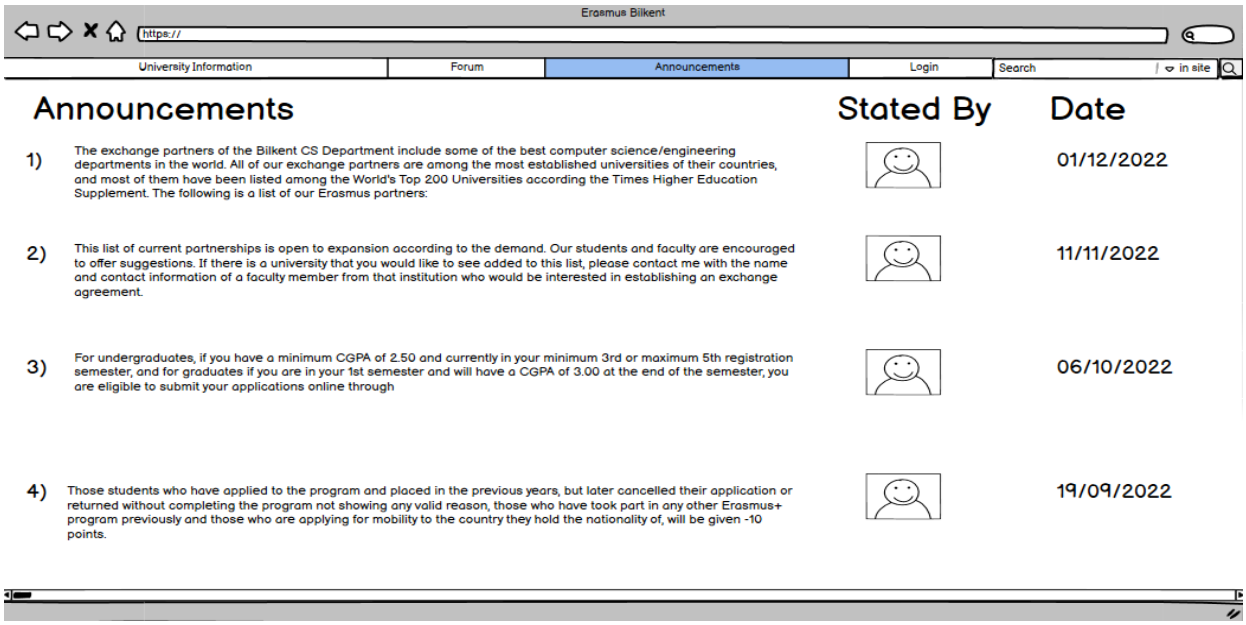

Figure 5.4.1 Forum Mock-up

This page will show the announcements, the person who stated them, and the announcement date.

5.5 Login

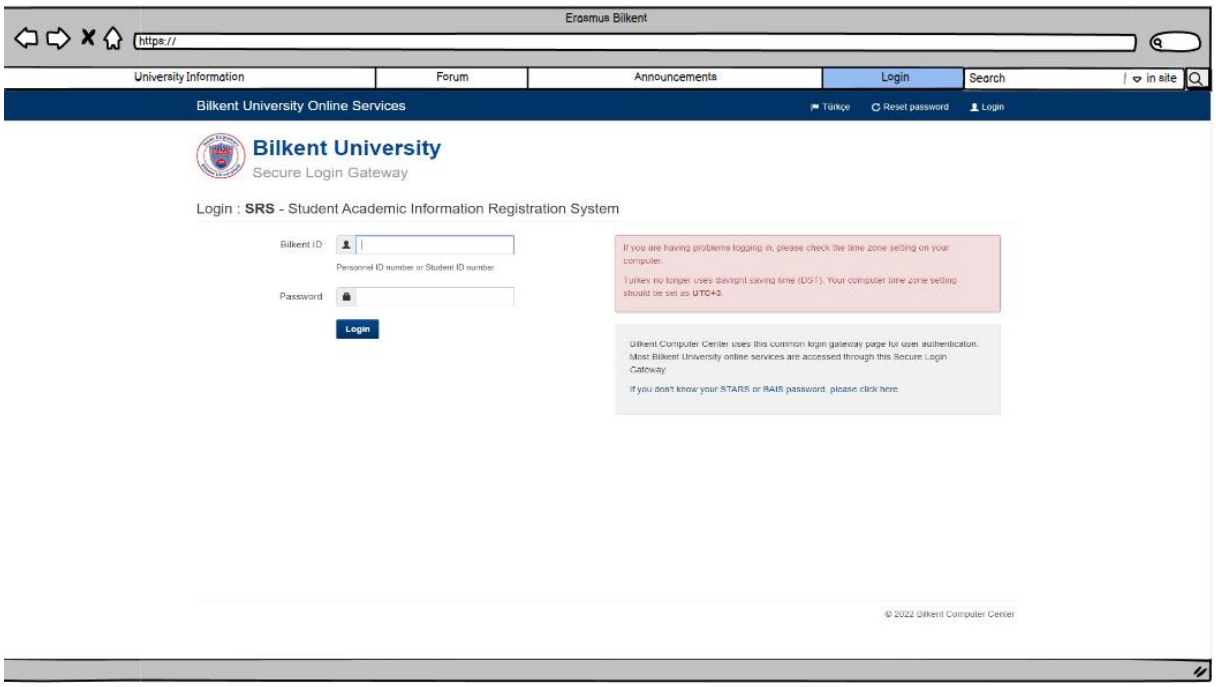

Figure 5.5.1 Login Page Mock-up

This is the login page. Since the webpage allows users to log in through SRS, the SRS login page will be shown.

5.6 Student (applying) Profile

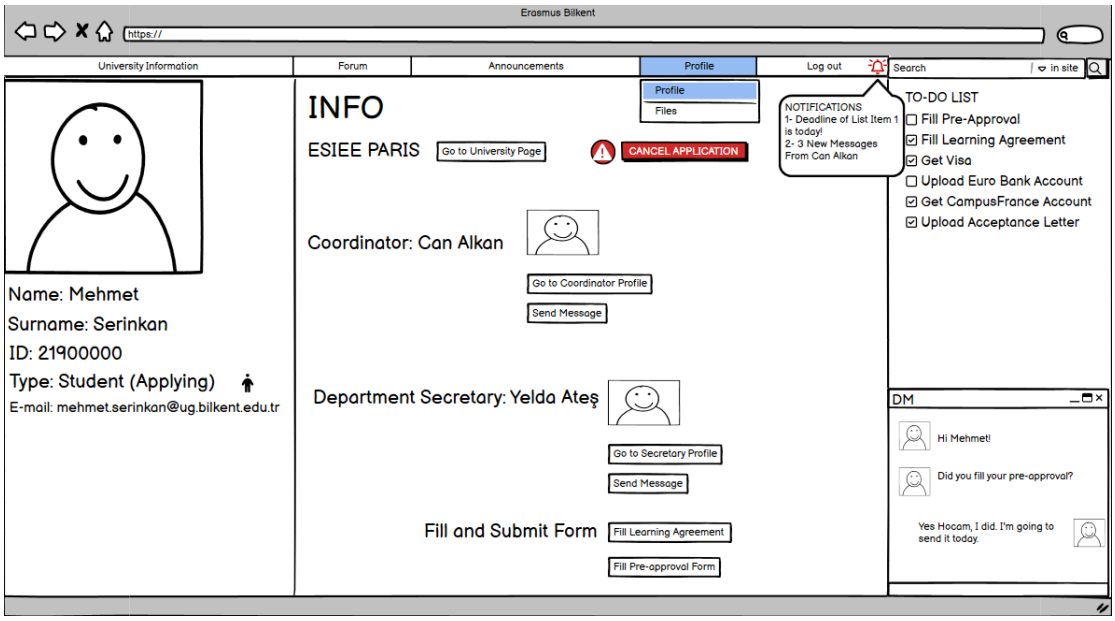

Figure 5.6.1 Profile1 Mock-up

When a student (applying) login, the profile will be opened. On the left part, the personal information will be shown including name, surname, ID, type, and email. On the middle part of the profile page, the general info and related users will be shown. User can also cancel the ERASMUS application, and he/she can go to fill the learning agreement and pre-approval form page by buttons. On the right, there will be To-Do List. Items will be shown one under the other. And there will be DM page. At the top, the notifications and a link to files page will be shown.

5.7 Files (student applying)

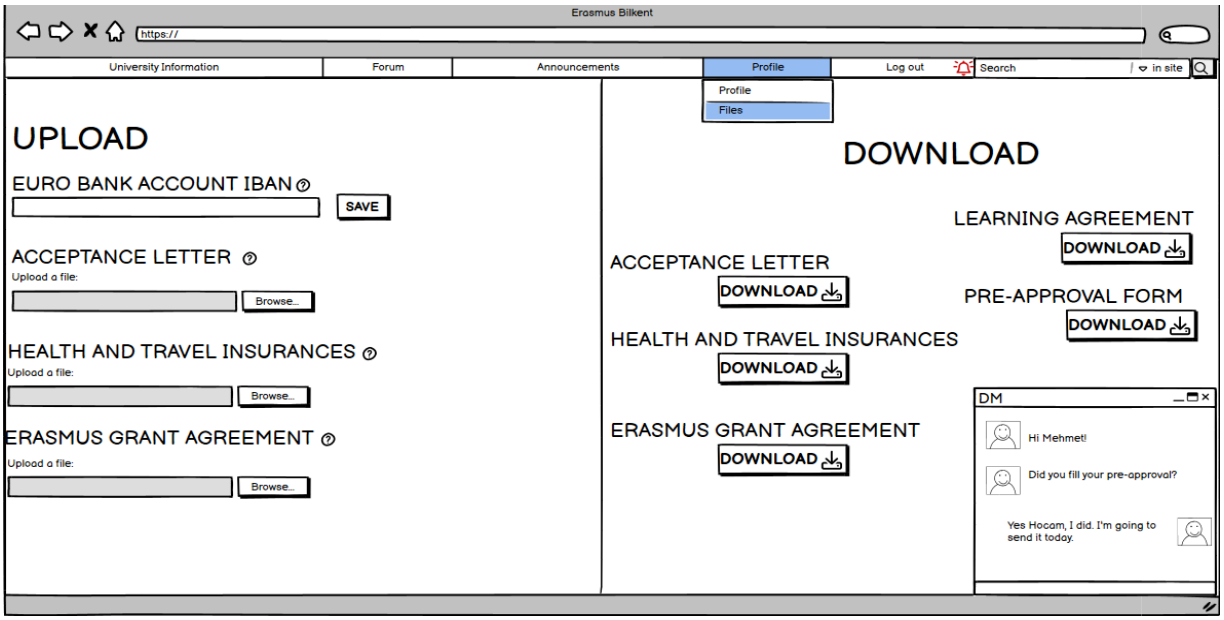

Figure 5.7.1 Files Page Mock-up

This is the files page. On the left part, the user can add the IBAN of the Euro Bank account, and he/she can upload the acceptance letter, health and travel insurance, ERASMUS Grant Agreement. On the right, the user can download these files, also the learning agreement and pre-approval form.

5.8 Fill and Submit Learning Agreement

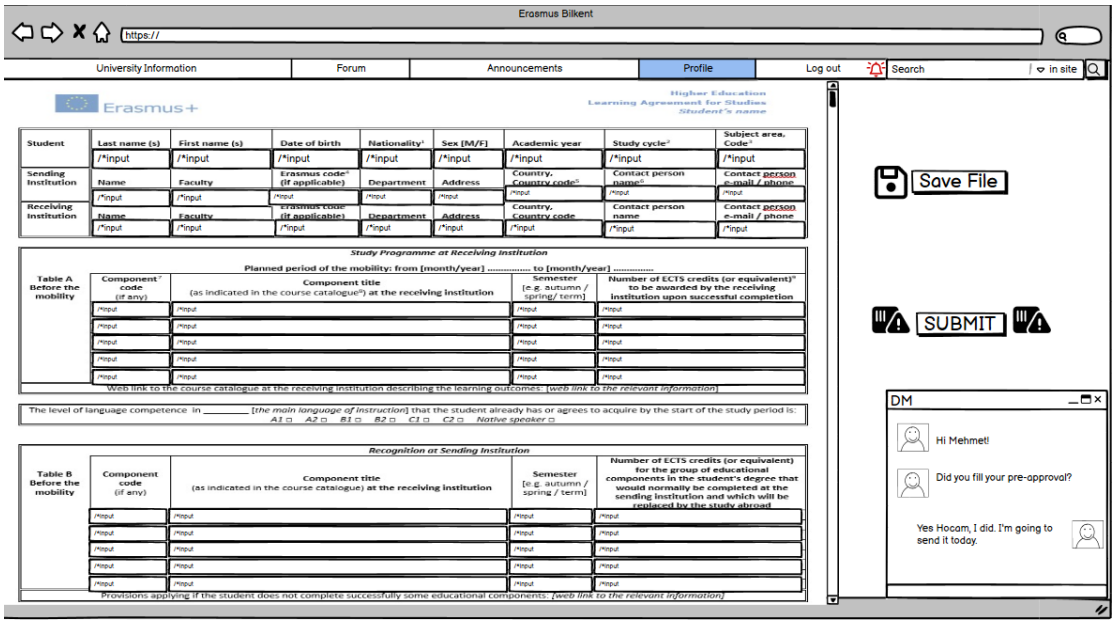

Figure 5.8.1 Form1 Mock-up

User can fill and submit the learning agreement. Also, if it has not finished yet, user can save the file to complete it later. There will be input fields to send backend API. These inputs will be saved onto backend. After that, when user uses this page, they will be loaded from backend.

5.9 Fill and Submit Pre-approval Form

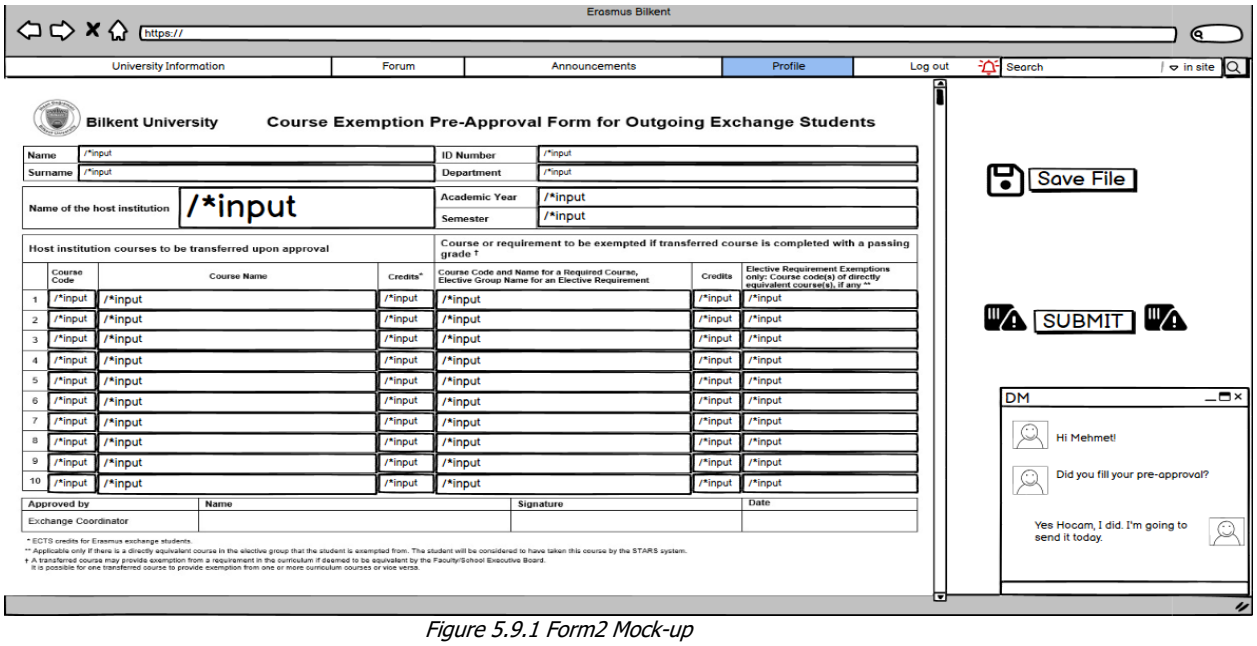

Figure 5.9.1 Form2 Mock-up

User can fill and submit the pre-approval. If it has not finished yet, user can save the file to complete it later. There will be input fields to send backend API. These inputs will be saved onto backend. After that, when user uses this page, they will be loaded from backend.

5.10 Student (former) Profile

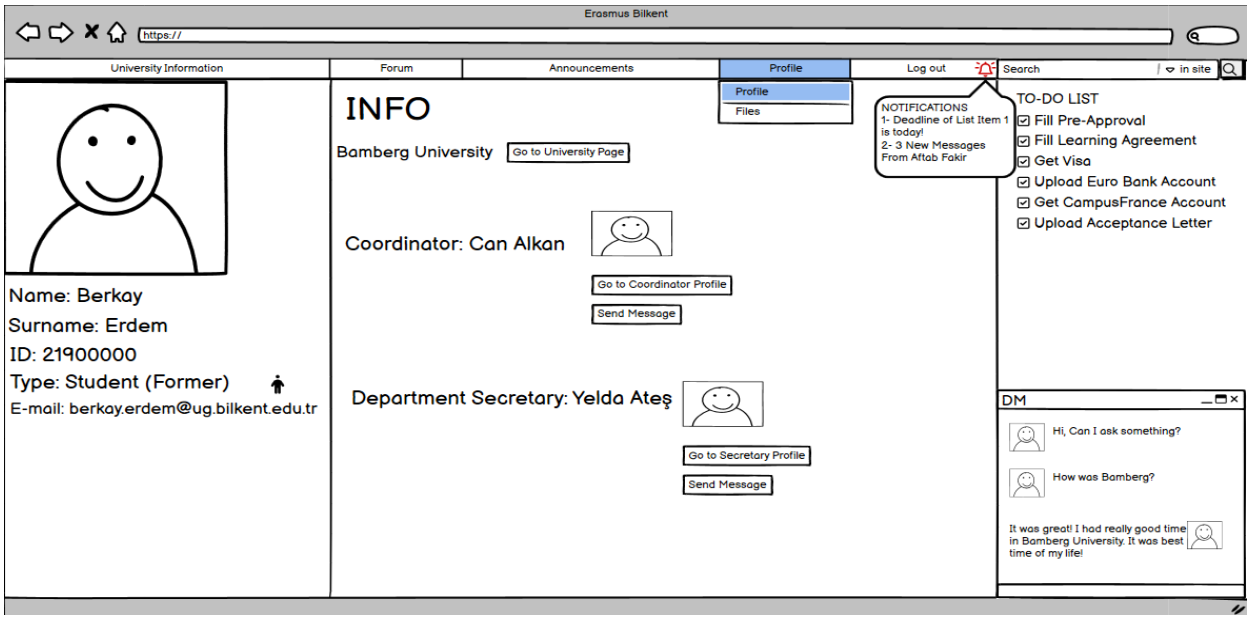

Figure 5.10.1 Profile2 Mock-up

The organization of former student profile is similar to applying student. On the left part, personal information will be shown. In the middle of the page, the general info and related users will be shown. On the right, there will be To-Do List. Items will be shown one under the other. And there will be DM page. At the top, the notifications and a link to files page will be shown.

5.11 Review Your University (Former student)

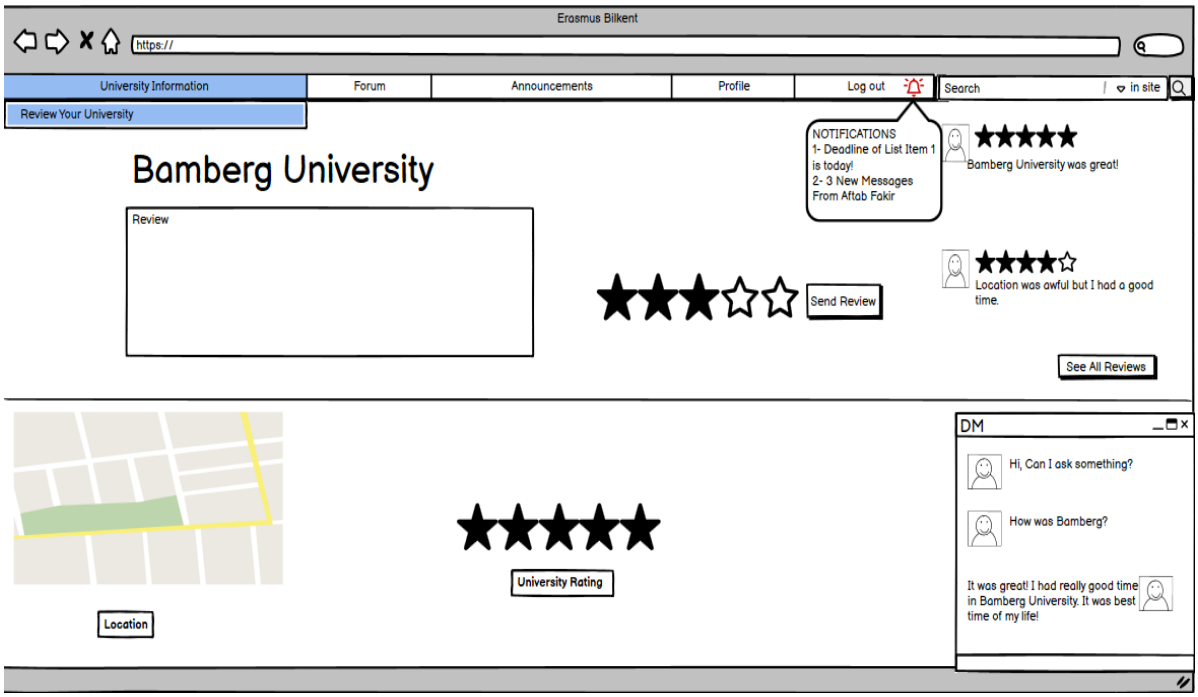

Figure 5.11.1 Review Page Mock-up

The page of reviewing university is consisted of review part, see all reviews of the university, location, rating and the points.

5.12 Exchange Office Screen

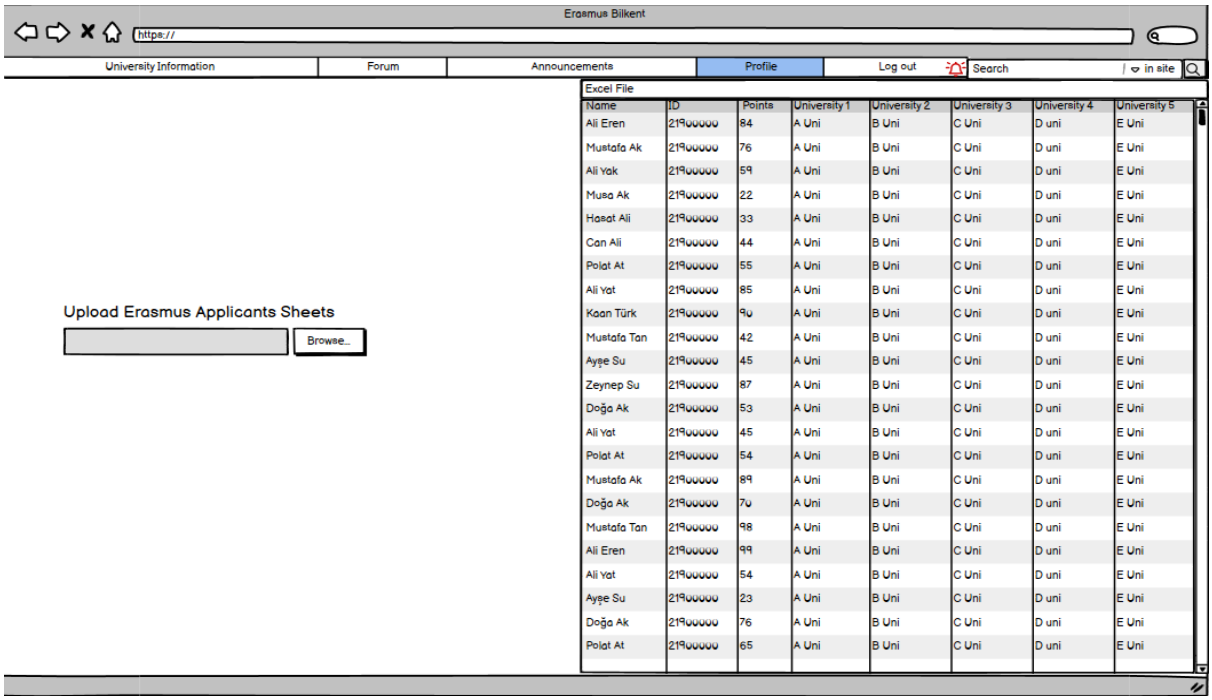

Figure 5.12.1 ExOffice Mock-up

Since the only assignment of the Erasmus Exchange Office is uploading the Erasmus Applicants Sheets, there will be only an upload file part. When this excel file is uploaded, it will be shown on the right part of the page.

5.13 Instructor Profile

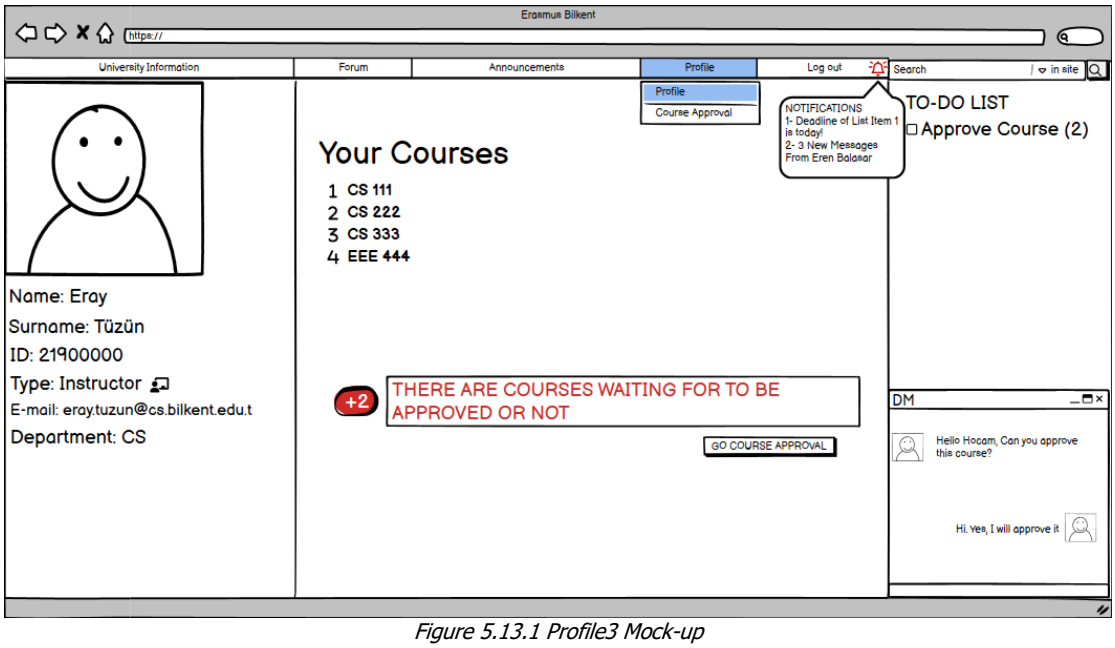

Figure 5.13.1 Profile3 Mock-up

On the left part, personal information will be shown. In the middle of the page, the courses of the instructor and the notifications of coursers waiting for to be approved or not will be shown. The instructor can go to the course approval page by using the button. Lastly, there will be a to-do list of the instructor and a DM panel.

5.14 Course Approval

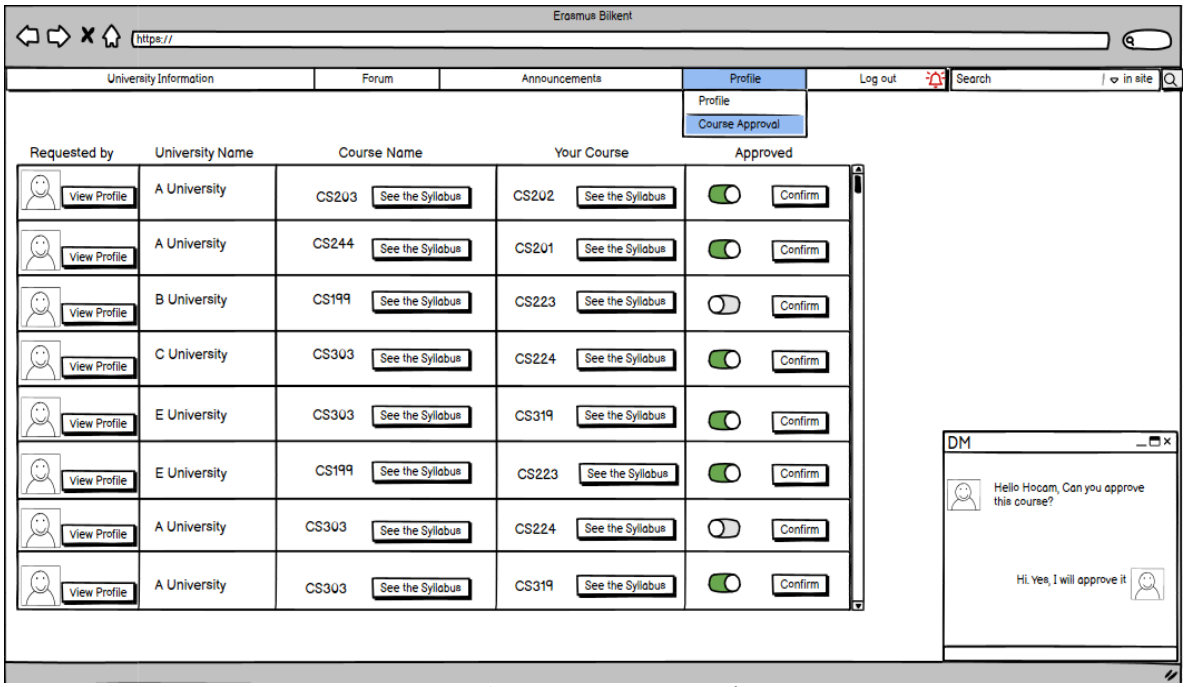

Figure 5.14.1 Course Approvai Page Mock-up

On the course approval page, there will be a list consisting of who requested, the student's university, the course name in the host university, the corresponding course, and the check of whether the instructor approved. When it is done, the instructor can confirm it by using button.

5.15 Profile View by Other Users

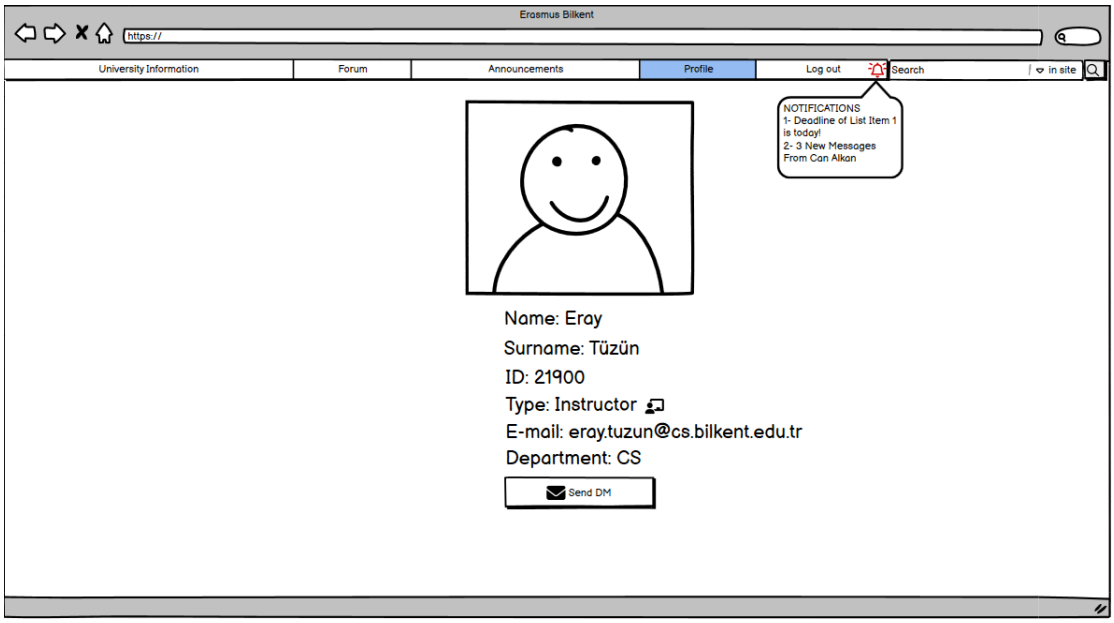

Figure 5.15.1 Profile4 Mock-up

When other typed users view the instructor profile, this page will be opened. The information of the instructor will be shown. Under them, there will be a send DM button. User can send a message with this button.

5.16 Department Coordinator Profile

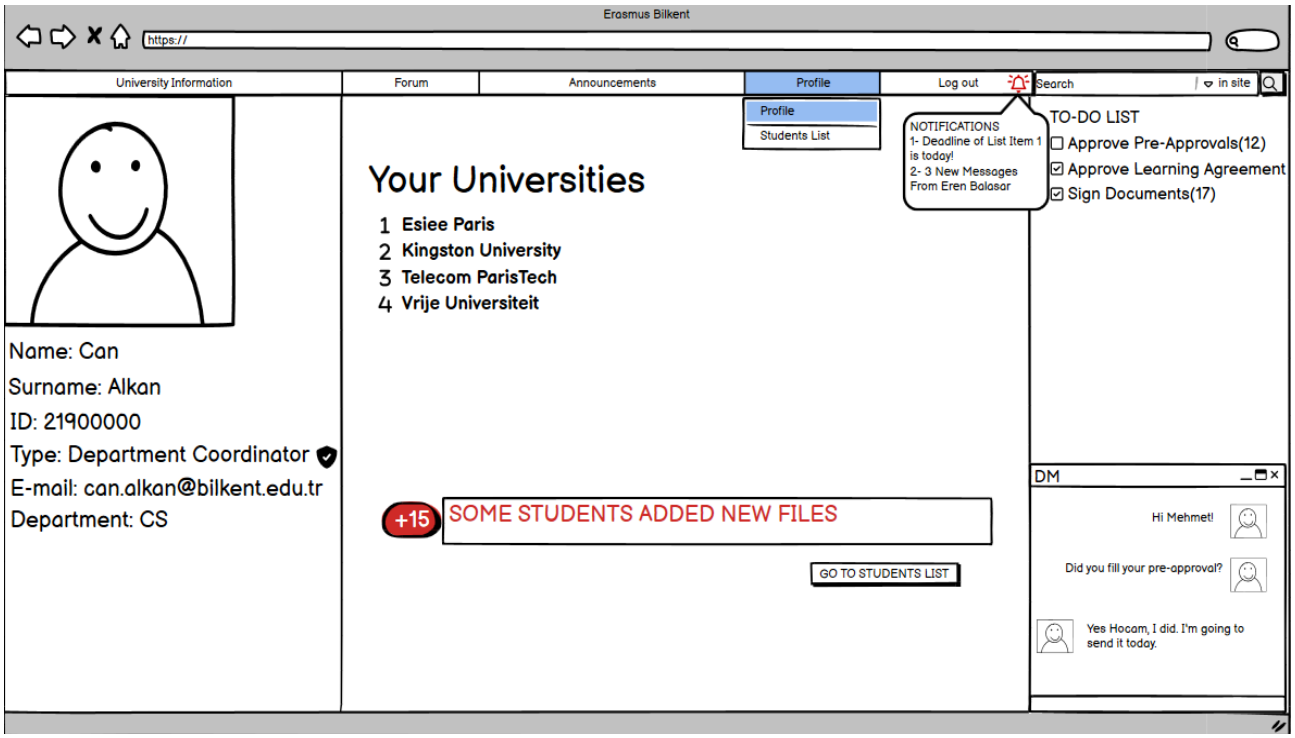

Figure 5.16.1 Profile5 Mock-up

There will be personal information on the left part of the website. The university department coordinator is responsible will be shown in the middle position. Also, the notifications of some students added new files will be shown. The left part of the notification indicates how many files have been added. By using the go to students list button, the department coordinator can go to the students list page. On the right part, the To-Do list and DM panel will be shown.

5.17 Student List

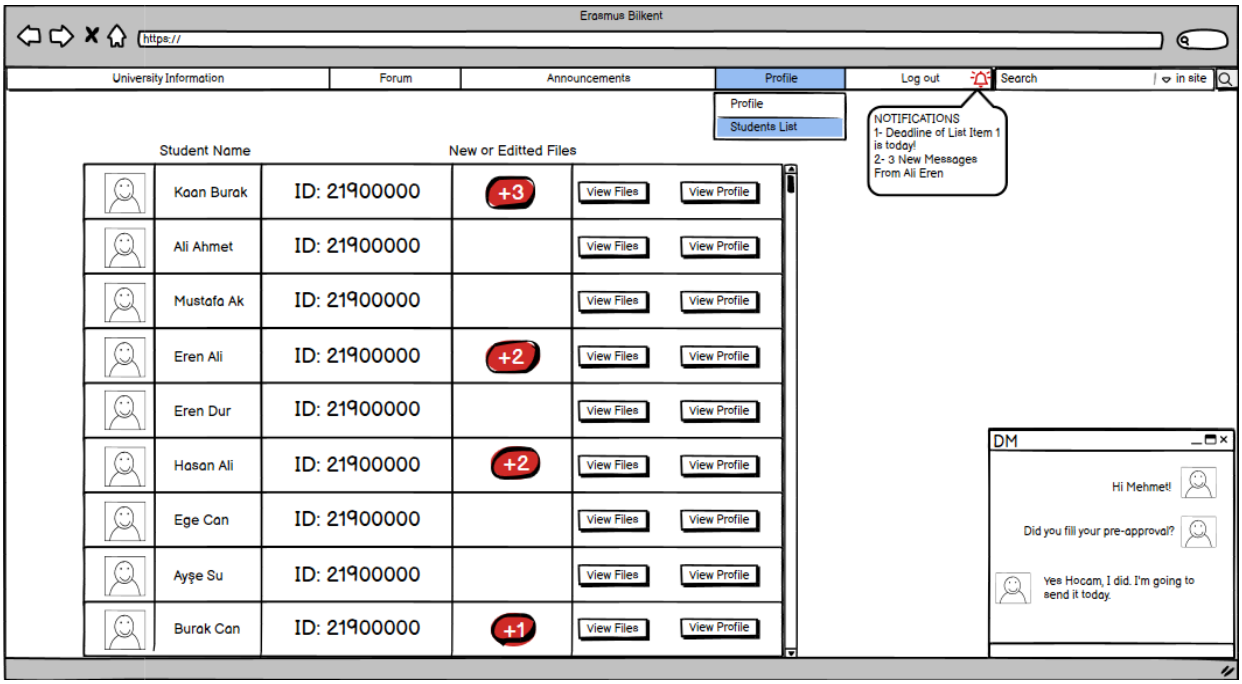

Figure 5.17.1 List Mock-up

There will be a list consisting of the student profile photo, student name, student ID, and/or edited files, view files, and view the student profile. By clicking the buttons view files and view profile, the department coordinator can go to the files of the student and the profile of the student.

5.18 Choosing Students

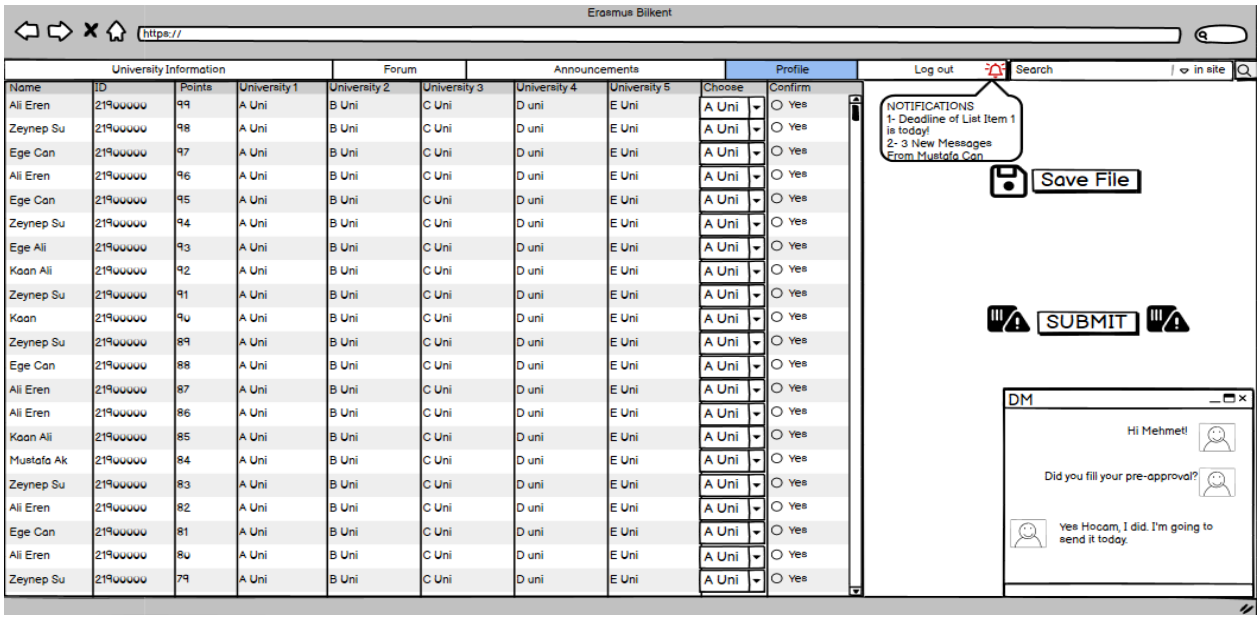

Figure 5.18.1 Choose Students Mock-up

The department coordinator can check the student applications. The list consists of name, id, points, preferred universities, and placed universities. At the end, the department coordinator can confirm by selecting the checkbox yes. He/she can submit it through Submit button, and it can be saved in order to be completed later by clicking the Save File button.

5.19 Exchange Coordinator Profile

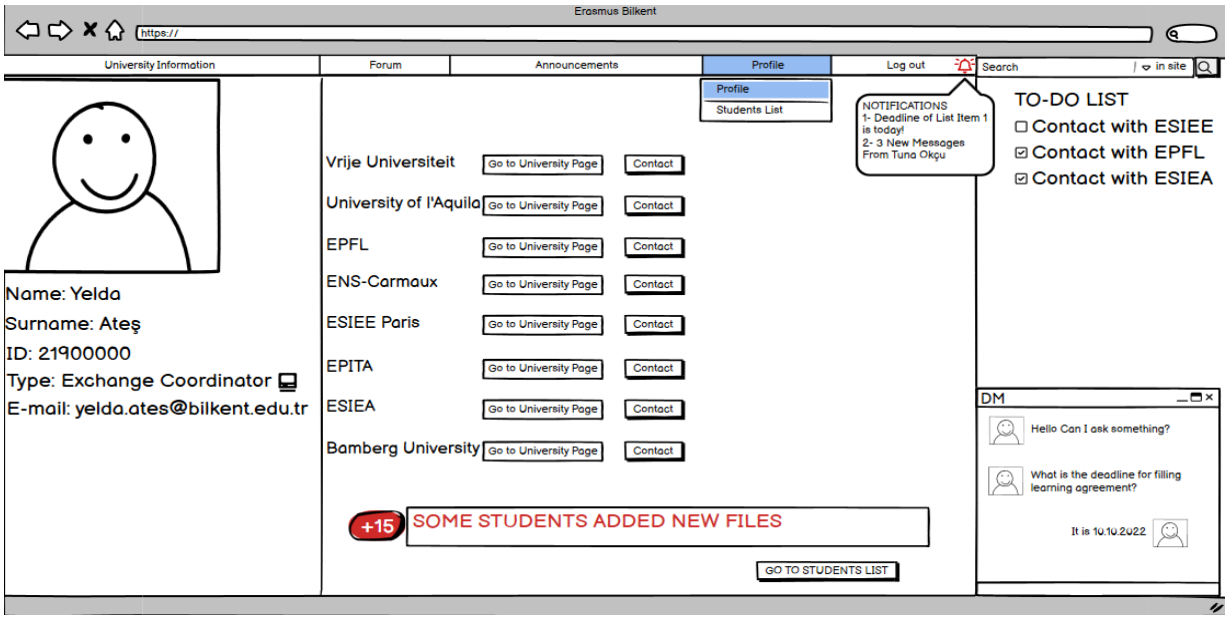

Figure 5.19.1 Profile 6 Mock-up

On the left, there will be personal information. In the middle of the page, there will be buttons for contacting and going to the university page. Moreover, there will be a notifications part that informs there are new files. The Department secretary can check new files by clicking the students' list button. It will open the students' list page. Lastly, on the right of the page, there will be a To-Do list and a DM panel.

6 Glossary

Mobility: The period starting from the moment a student goes to the host university and ending when he/she returns.

Exemption: If you have an exemption for a course, that means you will not have to take the corresponding course when you return from mobility.

7 References

[1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.