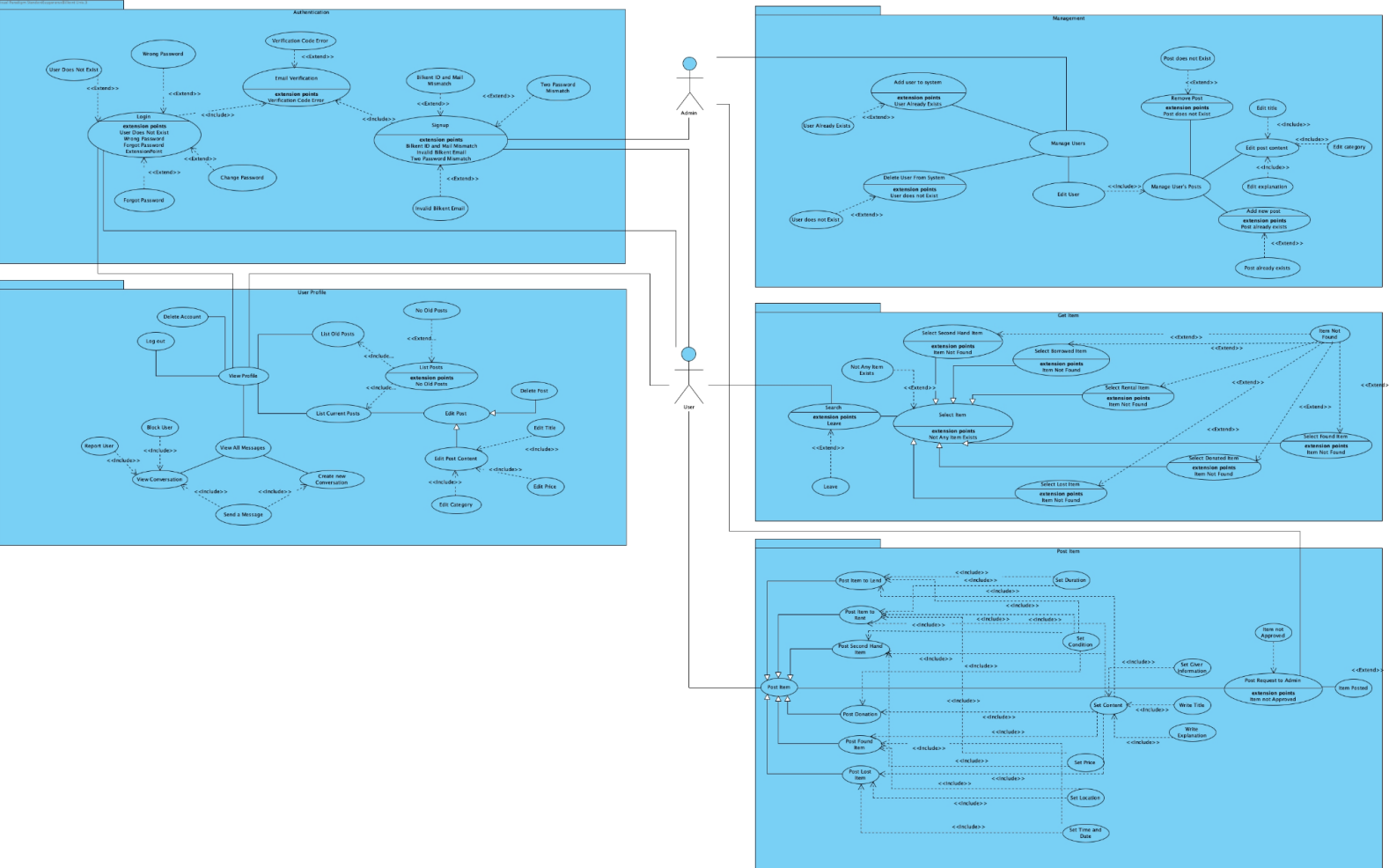


## D1- Use Case / NFR / Tech Stack (October 15 23:59)

### 1. Use Case Diagram of Campus Connect

#### 1.1. Use Case Diagram Screenshot



#### 1.2. Use Case Diagram Textual Descriptions

##### 1.2.1. Authentication Package

###### 1.2.1.1. Use Case Name: User does not exist

**Participating Actor:** User

**Entry Condition:** Mismatch during user login

**Exit Condition:** -

**Flow of Events:**

1. A user opens app and tries to log in
2. Given information does not match with users registered in the database

**Special / Quality Requirements:-**

###### 1.2.1.2. Use Case Name: Wrong Password

**Participating Actor:** User

**Entry Condition:** Mismatch during user login regarding password

**Exit Condition:** Password is entered correct

**Flow of Events:**

1. A user opens app and tries to log in
2. Given password information does not match with user that is registered in the database
3. Warning form appears

**Special / Quality Requirements: -**

**1.2.1.3. Use Case Name:** Forgot Password

**Participating Actor:** User

**Entry Condition:** User presses button notifying the system that login cannot be completed as user cannot remember the password for his account registered in the database

**Exit Condition: -**

**Flow of Events:**

1. User opens app and tries to log in
2. Despite there being an existing account user cannot provide password
3. User presses "Forgot Password" button
4. Form to reset and set new password opens
5. Information is updated on database with provided data

**Special / Quality Requirements: -**

**1.2.1.4. Use Case Name:** Change password

**Participating Actor:** User

**Entry Condition:** User wishes to change password and presses button to change during login page

**Exit Condition: -**

**Flow of Events:**

1. A user opens app and tries to log in
2. User wants to change password
3. The "Change Password" button is pressed
4. Form to enter profile information for which the password will be changed appears
5. Users logs into profile temporarily to enter new password
6. Information is updated on database

**Special / Quality Requirements:** Previous password should be entered correct

**1.2.1.5. Use Case Name:** Login

**Participating Actor:** User

**Entry Condition:** User opens app and this is the main page to welcome visitor

**Exit Condition: -**

**Flow of Events:**

The app is opened

**Special / Quality Requirements:-**

- 1.2.1.6. Use Case Name:** Verification Code Error  
**Participating Actor:** User  
**Entry Condition:** Login or signup step is completed and user needs to verify their email in order to complete the second step of authentication for strengthened security.  
**Exit Condition:** -  
**Flow of Events:**
1. A user completes login or signup
  2. User is directed to a new form where email verification will be completed via a specific digit code sent to users bilkent mail

**Special / Quality Requirements:-**

- 1.2.1.7. Use Case Name:** Verification code error  
**Participating Actor:** User  
**Entry Condition:** After email verification was sent, user fails to provide required information to the server  
**Exit Condition:** User either enters correct code or requests new code  
**Flow of Events:**
1. User completes login or signup
  2. User is directed to a new form where email verification will be completed via a specific digit code sent to users bilkent mail
  3. The code is either not entered in required time period or a wrong code is entered
  4. Warning message is displayed
- Special / Quality Requirements:** Imputed digit amount should not exceed required amount

- 1.2.1.8. Use Case Name:** Bilkent ID and mail mismatch  
**Participating Actor:** Admin  
**Entry Condition:** When new user is completing signup steps, entered bilkent mail does not match with the bilkent member  
**Exit Condition:** User is required to input signup steps again until match occurs  
**Flow of Events:**
1. User is signing up
  2. A bilkent mail and ID is provided by user
  3. Database returns false value for comparing the mail of bilkent members ID

**Special / Quality Requirements:-**

- 1.2.1.9. Use Case Name:** Two password mismatch  
**Participating Actor:** Admin

**Entry Condition:** During signup steps, the user is required to enter the password of their choosing twice. If these two do not match 1-1, the user needs to provide information for signup once more.

**Exit Condition:** Passwords match

**Flow of Events:**

1. User is signing up
2. Once the form requesting necessary information is filled and submitted system checks for password match
3. If passwords do not match warning message is displayed encouraging the user to fill out the form with requested parameters

**Special / Quality Requirements:-**

**1.2.1.10. Use Case Name:** Invalid bilkent email

**Participating Actor:** Admin

**Entry Condition:** After sign in form is filled, bilkent email does not correspond to valid address

**Exit Condition:** User either enters valid mail address

**Flow of Events:**

1. User is signing up
2. Once the form requesting necessary information is filled and submitted system checks for mail address validation
4. If the provided email address does not belong to any bilkent member, warning message is displayed

**Special / Quality Requirements:** Email should have bilkent address specifications

**1.2.1.11. Use Case Name:** Signup

**Participating Actor:** Admin

**Entry Condition:** User opens app and this is the main page to welcome visitor

**Exit Condition:** -

**Flow of Events:**

The app is opened

**Special / Quality Requirements:** -

## **1.2.2. User Profile Package**

**1.2.2.1. Use Case Name:** View All Messages

**Participating Actor:** User

**Entry Condition:**

- User is logged in
- User is viewing his/her profile

**Exit Condition:** -

**Flow of Events:**

1. The user accesses the "View All Messages" feature from their profile page
2. The system retrieves and displays all messages for the user

**Special / Quality Requirements:** The system should not overload all the messages. Users should be easily switch and select between them

**1.2.2.2. Use Case Name:** Create a New Conversation

**Participating Actor:** User

**Entry Condition:**

- The user is logged in
- The user has just used View All Messages and still using it

**Exit Condition:**

- Message is sent
- Message fails
- Message is canceled

**Flow of Events:**

1. The user composes a new message
2. The user selects a recipient
3. The user approves sending the message
4. The conversation is initiated

**Special / Quality Requirements:**

- Messages should not be more than 300 characters
- Message is transmitted without any delay

**1.2.2.3. Use Case Name:** View a Conversation

**Participating Actor:** User

**Entry Condition:**

- The user is logged in
- The user selects a valid conversation
- The user accesses the "View All Messages" feature from their profile page only

**Exit Condition:** Conversation pops-up.

**Flow of Events:**

1. The user accesses a conversation from their list
2. The system displays the conversation history
3. The design shows an option to close the chat

**Special / Quality Requirements:**

- Conversations should load quickly

**1.2.2.4. Use Case Name:** Send a Message

**Participating Actor:** User

**Entry Condition:**

- The user is in an active conversation.

**Exit Condition:** Message is sent.

**Flow of Events:**

1. The user inputs their message
2. The user confirms the message

**Special / Quality Requirements:**

- Messages have a maximum size of 300 characters

**1.2.2.5. Use Case Name:** Report User

**Participating Actor:** User

**Entry Condition:**

- The user is in an active conversation.
- The other user is valid

**Exit Condition:** The user confirms reporting the other user and the other user is reported

**Flow of Events:**

1. The user reports another user within a conversation.
2. The user confirms reporting the other user

**Special / Quality Requirements:**

- Reporting should be anonymous and handled confidentially.

**1.2.2.6. Use Case Name:** Block User

**Participating Actor:** User

**Entry Condition:**

- The user is in an active conversation with the target user.

**Exit Condition:** The user confirms blocking, and the target is blocked.

**Flow of Events:**

1. The user initiates blocking action
2. The user reports the cause of the blocking
3. The user confirms blocking

**Special / Quality Requirements:**

- Blocking should completely prevent the blocked user from further interaction

**1.2.2.7. Use Case Name:** Log Out

**Participating Actor:** User

**Entry Condition:**

- The user is logged in. (This also confirms that the user is a valid user.)
- User is in the profile page

**Exit Condition:** Return to sign in/register page

**Flow of Events:**

1. The user accesses the "View Profile" feature
2. User logs out

**Special / Quality Requirements:**

- The log out should be quick

**1.2.2.8. Use Case Name:** Delete Account

**Participating Actor:** User

**Entry Condition:**

- The user is logged in

- The user is in the profile page
- The user requests deletion of their account

**Exit Condition:** Return to the sign in/register page with all of the user information deleted and posts withdrawn

**Flow of Events:**

1. The user accesses the "View Profile" feature
2. User requests deletion of their account
3. All posted items are withdrawn
4. The account is deleted

**Special / Quality Requirements:**

- The deletion should be permanent and secure

#### 1.2.2.9. **Use Case Name:** View Profile

**Participating Actor:** User

**Entry Condition:**

- The user is logged in. (This also confirms that the user is a valid user.)

**Exit Condition:** Close request for the profile page

**Flow of Events:**

1. The user accesses the "View Profile" feature
2. The system displays the user's profile information

**Special / Quality Requirements:**

- The profile information should load quickly and compactly

#### 1.2.2.10. **Use Case Name:** List Old Posts

**Participating Actor:** User

**Entry Condition:**

- The user is logged in
- There are inactive items

**Exit Condition:** Items are listed

**Flow of Events:**

1. The user selects the "List Old Posts" option
2. The system displays a list of items old posts by the user

**Special / Quality Requirements:**

- The list should be compact, and the user should select to see more information

#### 1.2.2.11. **Use Case Name:** List Current Posts

**Participating Actor:** User

**Entry Condition:**

- The user is logged in
- The user has items that are active

**Exit Condition:** All posted items are listed

**Flow of Events:**

1. The user chooses the "List All Posted Items" feature
2. The system displays a list of active items posted by the user

**Special / Quality Requirements:**

- The list of listed items should load swiftly and resemble quick summaries

**1.2.2.12. Use Case Name:** List Posts

**Participating Actor:** User

**Entry Condition:**

- The user is viewing their listed items

**Exit Condition:** Designated items are listed

**Flow of Events:**

1. The user views the list of items
2. The items of the requested type (sold or posted) are displayed

**Special / Quality Requirements:**

- The list should be presented in an organized manner

**1.2.2.13. Use Case Name:** Edit Post

**Participating Actor:** User

**Entry Condition:**

- The user is logged in
- The user wants to edit an item listed for some or all of the attributes

**Exit Condition:** Editions are saved for one or more listed items

**Flow of Events:**

1. The user accesses the "Edit Post" feature
2. Required operation(s) is/are performed
3. Editions confirmed

**Special / Quality Requirements:**

- Has a user-friendly interface

**1.2.2.14. Use Case Name:** Delete Post

**Participating Actor:** User

**Entry Condition:**

- The user triggers "Edit Post"

**Exit Condition:** The item is deleted

**Flow of Events:**

1. The user selects an item to delete
2. The system confirms the deletion

**Special / Quality Requirements:**

- Deleted items cannot be recovered

**1.2.2.15. Use Case Name:** Edit Contents

**Participating Actor:** User



**Entry Condition:**

- The user triggers "Edit Post"

**Exit Condition:** The item attributes become editable

**Flow of Events:**

1. The user selects an item to edit

**Special / Quality Requirements:**

- Should not interrupt any marketplace operations

**1.2.2.16. Use Case Name:** Edit Title

**Participating Actor:** User

**Entry Condition:**

- The user is "Edit Post" state

**Exit Condition:** The item title is edited (includes null operation)

**Flow of Events:**

1. The user selects an item to edit
2. The user updates the title
3. The system saves the changes

**Special / Quality Requirements:**

- The name should change across all of the system

**1.2.2.17. Use Case Name:** Edit Category

**Participating Actor:** User

**Entry Condition:**

- The user is "Edit Post" state

**Exit Condition:** Item category is changed

**Flow of Events:**

1. The user selects an item to edit
2. The user updates the item's location
3. The system saves the changes

**Special / Quality Requirements:**

- Item's location should change across all of the systems

**1.2.2.18. Use Case Name:** Edit Price

**Participating Actor:** User

**Entry Condition:**

- The user is "Edit Post" state

**Exit Condition:** Price is edited

**Flow of Events:**

1. The user selects an item to edit
2. The user updates the item's price
3. The system saves the changes

**Special / Quality Requirements:**

- The price change should take effect swiftly

**1.2.2.19. Use Case Name:** No Old Post

**Participating Actor:** User

**Entry Condition:**

- The user is wants to view old posted items
- There are no sold items

**Exit Condition:** Display exception message

**Flow of Events:**

1. The user views their sold items
2. There are no items sold.

**Special / Quality Requirements:**

- The system recommends selling tips

### 1.2.3. Post Item Package

#### 1.2.3.1. Use Case Name: Post Item

**Participating Actor:** User

**Entry Condition:** User must be logged in

**Exit Condition:** -

**Flow of Events:**

1. Click to “Post Item” link at the navbar
2. Select item type to post

**Special / Quality Requirements:** -

#### 1.2.3.2. Use Case Name: Post Item to Lend

**Participating Actor:** User

**Entry Condition:** Selected item type must be type “Lended” at the “Post Item” stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements:** -

#### 1.2.3.3. Use Case Name: Post Item to Rent

**Participating Actor:** User

**Entry Condition:** Selected item type must be type “Rented” at the “Post Item” stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements:** -

#### 1.2.3.4. Use Case Name: Post Second Hand Item

**Participating Actor:** User

**Entry Condition:** Selected item type must be type “Second Hand” at the “Post Item” stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements: -**

**1.2.3.5. Use Case Name:** Post Donation

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Donation" at the "Post Item" stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements: -**

**1.2.3.6. Use Case Name:** Post Found Item

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Found" at the "Post Item" stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements: -**

**1.2.3.7. Use Case Name:** Post Lost Item

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Lost" at the "Post Item" stage

**Exit Condition:** -

**Flow of Events:**

1. Completely fill the information required about the item to be post to be able to post the item

**Special / Quality Requirements: -**

**1.2.3.8. Use Case Name:** Set Duration

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Lended" or "Rented" at the "Post Item" stage

**Exit Condition:** Duration set properly

**Flow of Events:**

1. Set the number of months to give item away
2. Set the number of days to give item away

**Special / Quality Requirements:** Duration can not be longer than 1 year

**1.2.3.9. Use Case Name:** Set Condition

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Lended", "Donation", "Second Hand" or "Rented" at the "Post Item" stage

**Exit Condition:** Condition set properly

**Flow of Events:**

1. Select the current condition/status of the item among these values: Like New, Very Good, Good, Fair, Poor

**Special / Quality Requirements:** -

**1.2.3.10. Use Case Name:** Set Content

**Participating Actor:** User

**Entry Condition:** User must be in "Post Item" stage

**Exit Condition:** Title, explanation and giver information set properly

**Flow of Events:**

1. Set requested information

**Special / Quality Requirements:** Title and explanation must not contain any bad words that admin would not approve

**1.2.3.11. Use Case Name:** Set Price

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Second Hand" or "Rented" at the "Post Item" stage

**Exit Condition:** Price set properly

**Flow of Events:**

1. Set price

**Special / Quality Requirements:** Price must be in a proper price range that admins would approve

**1.2.3.12. Use Case Name:** Set Location

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Lost" or "Found" at the "Post Item" stage

**Exit Condition:** Location set properly

**Flow of Events:**

1. Select proper location from the options available for users to select among

**Special / Quality Requirements:** -

**1.2.3.13. Use Case Name:** Set Time and Date

**Participating Actor:** User

**Entry Condition:** Selected item type must be type "Lost" or "Found" at the "Post Item" stage

**Exit Condition:** Time and Date set properly

**Flow of Events:**

1. Set approximate time the item found or lost as "hour:minute"
2. Set date

**Special / Quality Requirements:** Set time and date have to be past

- 1.2.3.14. Use Case Name:** Set Giver Information  
**Participating Actor:** User  
**Entry Condition:** User must be in “Post Item” stage  
**Exit Condition:** -  
**Flow of Events:**  
1. Select information while posting from the available information about yourself in your profile that you would like to be seen by everyone using the app  
**Special / Quality Requirements:** Name and surname information must be displayed to other users to make in-app communication more secure and proper
- 1.2.3.15. Use Case Name:** Write Title  
**Participating Actor:** User  
**Entry Condition:** User must be in “Post Item” stage  
**Exit Condition:** Title set properly  
**Flow of Events:**  
1. Write a compatible title for the item to be posted  
**Special / Quality Requirements:** Title should not include bad words
- 1.2.3.16. Use Case Name:** Write Explanation  
**Participating Actor:** User  
**Entry Condition:** User must be in “Post Item” stage  
**Exit Condition:** Explanation set properly  
**Flow of Events:**  
1. Write a brief explanation for the item about the things you would like to include  
**Special / Quality Requirements:** Explanation should not include bad words
- 1.2.3.17. Use Case Name:** Post Request to Admin  
**Participating Actor:** User  
**Entry Condition:** Information about the item must be filled without any defects  
**Exit Condition:** Post request sent to admin properly  
**Flow of Events:**  
1. Clicking to “Post Request to Admin” button at the “Post Item” page  
**Special / Quality Requirements:** -
- 1.2.3.18. Use Case Name:** Item not Approved  
**Participating Actor:** User, Admin  
**Entry Condition:** A request must be sent to post an item to admin and that request to post item is denied by the admin  
**Exit Condition:** -  
**Flow of Events:**

1. To the message box of the related user a default message saying something like “Your post item request has been denied by the application admins” is sent

**Special / Quality Requirements:** The process of viewing an item to decide whether approve the request from the user according to post the related item or deny should not exceed the time limit of 3 days

**1.2.3.19. Use Case Name:** Item Posted

**Participating Actor:** User

**Entry Condition:** A request must be sent to post an item to admin

**Exit Condition:** -

**Flow of Events:**

1. Item is posted to the site
2. Item is added to the “Posted Items” of the related user

**Special / Quality Requirements:** -

**1.2.4. Get Item Package**

**1.2.4.1. Use Case Name:** Search

**Participating Actor:** User

**Entry Condition:** The user enters from the login/sign-up screen.

**Exit Condition:** The user goes to the post-item screen or does a specific categorisation and goes to other use cases.

**Flow of Events:**

1. If the user wants to post any item, the user clicks the post item button on the screen because the search screen is also the main screen of our application.
2. If the user wants to do a specific search, which includes a categorisation, first they click the specific area that they want (second hand, borrowing, etc.)
3. Then they immediately go to a new use case.

**1.2.4.2. Use Case Name:** Select Item

**Participating Actor:** User

**Entry Condition:** The user must continue from the search screen without going to the Post Item Screen.

**Exit Condition:** The user must decide to leave the select item/search screen and get out of the application or decide to post an item

**Flow of Events:**

1. User must decide to select which type of item he/she wants.
2. After selecting the type of item he/she wants he goes to that use case and the selected item use case is the main use case that he/she stays in.

3. The user also can decide to cancel the selected item stage and goes back to the main screen of the application

**Special / Quality Requirements:**

This use case allows users to navigate to specific type of select item use cases easier basically for the sake of the applications work efficiency.

**1.2.4.3. Use Case Name:** Select Second Hand Item

**Participating Actor:** User

**Entry Condition:** User must press the “Second Hand Item” from the null bar.

**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case the user must select this type of item from the null bar.
2. After entering this use case user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants and there will be new options for the user after it such as communicating with the seller etc.

**1.2.4.4. Use Case Name:** Select Borrowed Item

**Participating Actor:** User

**Entry Condition:** User must select the appropriate option from the null bar for entering this state.

**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case the user must select this type of item from the null bar.
2. After entering this use case, the user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants and there will be new options for the user after it such as communicating with the seller etc.

**1.2.4.5. Use Case Name:** Select Rental Item

**Participating Actor:** User

**Entry Condition:** The user must select the appropriate option from the null bar for entering this state.

**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case, the user must select this type of item from the null bar.
2. After entering this use case, the user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants. There will be new options for the user after it, such as communicating with the seller, etc.

**1.2.4.6. Use Case Name:** Select Found Item

**Participating Actor:** User

**Entry Condition:** The user must select the appropriate option from the null bar for entering this state.

**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case, the user must select this type of item from the null bar.
2. After entering this use case, the user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants. There will be new options for the user after it, such as communicating with the seller, etc.

**1.2.4.7. Use Case Name:** Select Donated Item

**Participating Actor:** User

**Entry Condition:** The user must select the appropriate option from the null bar for entering this state.

**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case, the user must select this type of item from the null bar.
2. After entering this use case, the user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants. There will be new options for the user after it, such as communicating with the seller, etc.

**1.2.4.8. Use Case Name:** Select Lost Item

**Participating Actor:** User

**Entry Condition:** The user must select the appropriate option from the null bar for entering this state.



**Exit Condition:** The user may decide to select/buy a different type of item. Or even decide not to buy at all.

**Flow of Events:**

1. For entering this use case, the user must select this type of item from the null bar.
2. After entering this use case, the user can decide to leave and go back.
3. Another option for the user is to go and select the second-hand item he/she wants. There will be new options for the user after it, such as communicating with the seller, etc.

**1.2.4.9. Use Case Name:** Item Not Found

**Participating Actor:** User

**Entry Condition:** For entering this use case after entering specific select stages the item retrieval from the database should resolve with an unfound error.

**Exit Condition:** The only exit condition is to go back to the general search screen and look for other items because this item is already not found in the database and gives an error.

**Flow of Events:** There is no specific flow in this use case this is an “extends” case, so the user should just go back.

**Special / Quality Requirements:** This is a specific error-handling case.

**1.2.4.10. Use Case Name:** Not Any Item Exists

**Participating Actor:** User

**Entry Condition:** For entering this use case in the database of the application, there must be zero items, so this occurs when all of the posts are deleted or at the initial state of the application.

**Exit Condition:** An item should be posted.

**Flow of Events:** There is not a flow in this case either, because this is an “extends” case.

**Special / Quality Requirements:** This is a specific error-handling case.

**1.2.4.11. Use Case Name:** Leave

**Participating Actor:** User

**Entry Condition:** The user must decide to leave the application.

**Flow of Events:** The User presses the “leave” button of the application or just closes the web page.

**1.2.5. Management Package**

**1.2.5.1. Use Case Name:** Manage Users

**Participating Actor:** Admin

**Entry Condition:** Admin is logged in and selects this option.

**Exit Condition:** Admin decides to choose the following options.

**Flow of Events:** Admin chooses this option after logging in. He/she then can choose which part to change between the next 3 options.

**Special / Quality Requirements:** This case is to create a main page that offers the admin to his/her controls on the users.

**1.2.5.2. Use Case Name:** Edit User

**Participating Actor:** Admin

**Entry Condition:** Selected from “Manage Users” (1.2.5.1)

**Exit Condition:** Actor is done editing and quits the page, or continues to “Manage User’s Posts” option.

**Flow of Events:** The actor is directed to the “Manage User’s Posts” page of the requested ID numbered user to edit that user’s posts, if that option is chosen.

**Special / Quality Requirements:** Actor needs to select the ID number of the user to continue editing posts of a specific user.

**1.2.5.3. Use Case Name:** Delete User from System

**Participating Actor:** Admin

**Entry Condition:** Selected from “Manage Users” (1.2.5.1)

**Exit Condition:** Accomplishing deleting the user with given Bilkent ID, or skipping to “User does not Exist” (1.2.5.6), or quitting to delete the account.

**Flow of Events:** Actor selects which user account to delete from user IDs on the database. If no such user with the selected ID is found on the system database, the actor is directed to the “User does not Exist” page.

**Special / Quality Requirements:** The actor needs to select a valid and existing account that can be retrieved from the database.

**1.2.5.4. Use Case Name:** Add User to System

**Participating Actor:** Admin

**Entry Condition:** Selected from “Manage Users” (1.2.5.1)

**Exit Condition:** Accomplishing to add the new user, or skipping to “User Already Exists” (1.2.5.5).

**Flow of Events:** Actor directs the information of the new user to the database to create a new user account if no user with the given Bilkent ID exists.

**Special / Quality Requirements:** If a user with a given new user ID already exists and can be retrieved from the system database, the user can not accomplish adding that user again to the system.

**1.2.5.5. Use Case Name:** User Already Exists

**Participating Actor:** Admin

**Entry Condition:** In “Add User to System” (1.2.5.4), a user with the given ID number already has an account in the system.

**Exit Condition:** The actor has to go back to the previous page.

**Flow of Events:** This page is an error page that alerts the user that a new user can not be added since it already exists.

**Special / Quality Requirements:** User has no next options but going back to the previous page.

**1.2.5.6. Use Case Name:** User does not Exist

**Participating Actor:** Admin

**Entry Condition:** In “Delete User from System” (1.2.5.3), a user with the given ID number does not have an account in the system.

**Exit Condition:** The actor has to go back to the previous page.

**Flow of Events:** This page is an error page that alerts the user that a new user can not be added since it already exists.

**Special / Quality Requirements:** User has no next options but going back to the previous page.

**1.2.5.7. Use Case Name:** Manage User’s Posts

**Participating Actor:** Admin

**Entry Condition:** Selected from “Edit Users” (1.2.5.2)

**Exit Condition:** The actor quits the page and goes back to “Edit User” page to edit another user, or chooses among editing options for the current user with the requested ID.

**Flow of Events:** The actor chooses a user to edit from the user database, and is directed to this page that offers editing options.

**Special / Quality Requirements:** This page is only a manual to show and direct the actor to next editing pages.

**1.2.5.8. Use Case Name:** Remove Post

**Participating Actor:** Admin

**Entry Condition:** The actor chooses to remove a post of the user he/she is currently editing.

**Exit Condition:** The actor quits deleting and goes back to the previous page, or accomplishes removing a post of the user, or is directed to “Post does not Exist” if the user has no posts to delete.

**Flow of Events:**

1. The posts of the current user that the actor is editing are displayed.
2. Actor chooses among these posts which one to delete.

3. If the user does not have any posts, the actor is directed to the "Post does not Exist" page.

**Special / Quality Requirements:** The actor needs to make a choice between posts and, he/she can remove another one after accomplishing to remove a post.

**1.2.5.9. Use Case Name:** Edit Post Content

**Participating Actor:** Admin

**Entry Condition:** The actor chooses to edit a post of the user he/she is currently editing.

**Exit Condition:** The actor accomplishes to edit the selected post of the user he/she is currently editing, or quits and goes back to the previous page, or chooses which part of the post to edit and is directed to that page.

**Flow of Events:**

1. The actor chooses which post to edit among displayed posts of the user, which are retrieved from the database.
2. Actor is directed to the chosen page.

**Special / Quality Requirements:** This is a manual page to offer the actor which posts he/she can edit and then which feature they want to change.

**1.2.5.10. Use Case Name:** Edit Title

**Participating Actor:** Admin

**Entry Condition:** The actor chooses to edit the title of the previously chosen post.

**Exit Condition:** The actor completes the changes and goes back to the previous page, or quits editing.

**Flow of Events:** The actor changes the title of the post they have chosen.

**Special / Quality Requirements:** The actor can continue changing another post by going back to previous pages if they wish.

**1.2.5.11. Use Case Name:** Edit Category

**Participating Actor:** Admin

**Entry Condition:** The actor chooses to edit the category of the previously chosen post.

**Exit Condition:** The actor completes the changes and goes back to the previous page, or quits editing.

**Flow of Events:** The actor chooses which category that the item on the post actually belongs to among multiple choices of existing categories.

**Special / Quality Requirements:** The actor can continue changing another post by going back to previous pages if they wish.

- 1.2.5.12. Use Case Name:** Edit Explanations  
**Participating Actor:** Admin  
**Entry Condition:** The actor chooses to edit the category of the previously chosen post.  
**Exit Condition:** The actor completes the changes and goes back to the previous page, or quits editing.  
**Flow of Events:** The actor changes the title of the post they have chosen.  
**Special / Quality Requirements:** The actor can continue changing another post by going back to previous pages if they wish.
- 1.2.5.13. Use Case Name:** Add new Post  
**Participating Actor:** Admin  
**Entry Condition:** The actor chooses this page to add a new post to a user.  
**Exit Condition:** A post is added, or the actor quits the page and goes back to the previous page "Manage User's Posts", or is directed to "Post Already Exists".  
**Flow of Events:**  
  1. The information of a new post is taken from the user.
  2. The admin creates a new post item with given information and assigns it as a post of the user that gave the post information with a new Post ID.
  3. If a post with the same information exists, the actor is directed to the "Post Already Exists page".
  4. The information of the new post and its owner user is directed to the database.**Special / Quality Requirements:** This action is performed after a user wants to add a new post.
- 1.2.5.14. Use Case Name:** Post Already Exists  
**Participating Actor:** Admin  
**Entry Condition:** A post with the same information already exists on the "Add new Post" page , the actor is directed to the "Post Already Exists" page.  
**Exit Condition:** The actor must go back to the previous page.  
**Flow of Events:** The actor is alerted that an identical post exists.  
**Special / Quality Requirements:** User has no next options but going back to the previous page.
- 1.2.5.15. Use Case Name:** Post does not Exist  
**Participating Actor:** Admin  
**Entry Condition:** The user with the given User ID on the "Remove Post" page does not have any posts on the system database, the actor is directed to the "Post does not Exists" page.

**Exit Condition:** The actor must go back to the previous page.

**Flow of Events:** The actor is alerted that the requested post to be removed does not exist.

**Special / Quality Requirements:** User has no next options but going back to the previous page.

## **2. Non-Functional Requirements (NFR)**

### **2.1. User-Friendly Interface**

This application provides a detailed, glamorous, and user-friendly interface. Our objective was to merge these 3 makings in the same application. We designed our web application to be as minimal as possible for ease of the user. Our goal is to present the web app's elements in a polished and tidy manner, aiming for a clean and straightforward appearance. Every component should be easily accessible, ensuring a user-friendly experience for a more enjoyable interaction.

### **2.2. Safety**

Whilst designing our application, we added some features that will protect the user's privacy to ensure that using this application does not bring any safety/security problems.

### **2.3. Maintainability**

The student sales platform at Bilkent University is designed with maintainability in mind. Utilizing Object-Oriented Programming principles ensures a modular and adaptable structure, simplifying the addition of new features or quick bug fixes. This approach safeguards the main codebase from disruptions during optimisations, ensuring smooth and efficient maintenance.

### **2.4. User Satisfaction**

Without a user, our application is useless the more users interact with our application, the better it is for us because more users mean more posts, and more posts mean more interaction inside the application, so user satisfaction is one of our main non-functional requirements whilst designing this web-application.

## **3. Tech Stack**

We will use React.js for the frontend part of our application, Java Spring for the backend and databases and SQL, will use PostgreSQL.