USE CASE MODEL:

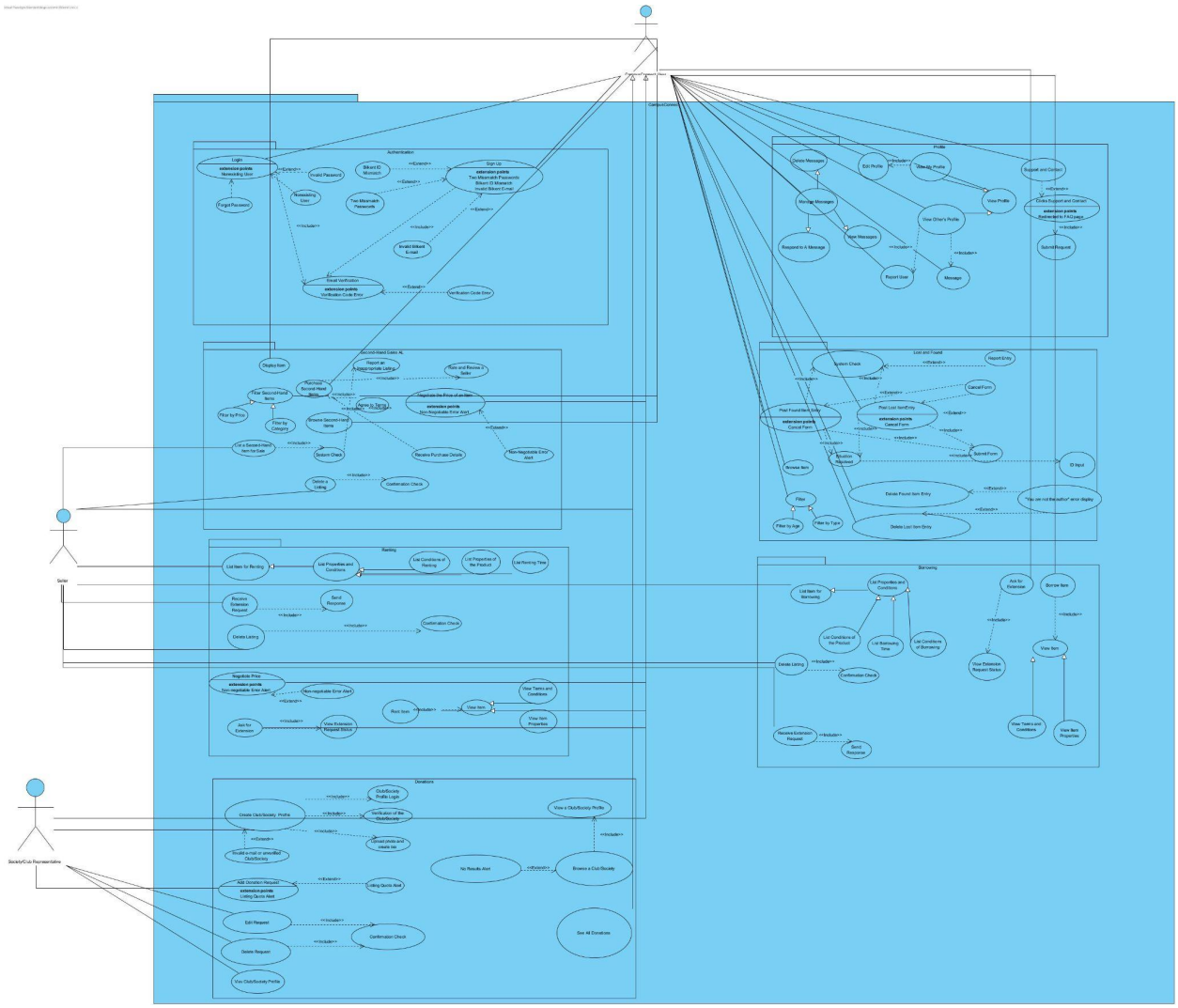

Figure 1: Use Case Model of CampusConnect

Second-Hand Sales Package

1. **Name:** Filter Second-Hand Items
2. **Participating Actors:** CampusConnect User
3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User is shown a list of items according to the filters applied.
5. **Flow of Events:**
   5.1. User clicks on the filter icon.
   5.2. User chooses a category (e.g., department, housing, clothing) and/or price filters.
   5.3. User presses apply.
   5.4. The system displays a list of relevant second-hand items with their details and prices.

1. **Name:** List a Second-Hand Item for Sale
2. **Participating Actors:** Registered Sellers
3. **Entry Conditions:** Seller is logged in.
4. **Exit Conditions:** Item is successfully listed for sale or the listing process is canceled.
5. **Flow of Events:**
   5.1. Seller selects the "List an Item for Sale" option.
   5.2. Seller provides detailed information about the item, including category, price, condition, and description.
   5.3. Seller uploads photos of the item.
   5.4. Seller chooses whether the listing is negotiable or non-negotiable.
   5.5. Seller confirms the listing details and posts the listing.
   5.6. The system verifies the listing and makes it visible to potential buyers.
6. **Special/Quality Requirements:** The system should provide accurate and up-to-date item listings.

1. **Name:** Report an Inappropriate Listing
2. **Participating Actors:** CampusConnect Users
3. **Entry Conditions:** User encounters a listing that violates platform guidelines.
4. **Exit Conditions:** The reported listing is reviewed by platform administrators.
5. **Flow of Events:**
   5.1. User identifies an inappropriate or suspicious listing.
   5.2. User selects the option to report the listing.
   5.3. User provides details of the issue and any supporting information.
   5.4. The system records the report and notifies platform administrators.
   5.5. Administrators review the report and take appropriate action, which may include removing the listing.
6. **Special/Quality Requirements:** The system should have mechanisms to handle and address reported issues promptly.

1. **Name:** Browse Second-Hand Items
2. **Participating Actors:** CampusConnect Users

3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User is shown the items related to the search query.
5. **Flow of Events:**
   5.1. User types in the product that they are interested in in the search bar.
   5.2. User clicks the “Search” button.
   5.2. The system displays a list of relevant second-hand items with their details and prices.

1. **Name:** Rate and Review a Seller
2. **Participating Actors:** CampusConnect Users
3. **Entry Conditions:** User has purchased an item.
4. **Exit Conditions:** User submits a rating and review for the seller.
5. **Flow of Events:**
   5.1. User goes to their purchase history.
   5.2. User selects the completed transaction, rates the seller on a scale, and leaves a written review.
   5.3. The system records the rating and review for the seller's profile.
6. **Special/Quality Requirements:** The system should ensure that reviews are genuine and provide a platform for seller feedback.

1. **Name:** Purchase Second-Hand Items
2. **Participating Actors:** CampusConnect Users, Registered Sellers
3. **Entry Conditions:** User is interested in a listed item and wants to make a purchase.
4. **Exit Conditions:** The purchase is completed, or the user cancels the purchase.
5. **Flow of Events:**
   5.1. User selects an item and clicks the "Buy Now" button.
   5.2. User confirms the purchase, selects the payment method, and provides necessary payment details.
   5.3. Seller receives a notification of the purchase.
   5.4. Seller confirms the sale and agrees to the terms.
   5.5. Payment is processed and the item is marked as sold.
   5.6. User and seller perceive purchase confirmation details.
6. **Special/Quality Requirements:** The system should ensure secure payment processing and order confirmation.

1. **Name:** Delete a Listing
2. **Participating Actors:** Registered Sellers
3. **Entry Condition:** Seller has listed an item for sale.
4. **Exit Condition:** The listing is deleted from the system.
5. **Flow of Events:**
   5.1. Seller goes to their listings.
   5.2. Seller selects the listing they want to delete.
   5.3. Seller clicks on the “Delete Listing” button, located at the bottom of the page.
   5.4. The system sends a prompt to confirm that the seller wants to delete the listing.

6. **Special/Quality Requirements:** The item must not be sold.

1. **Name:** Negotiate the Price of an Item
2. **Participating Actors:** CampusConnect Users, Registered Sellers
3. **Entry Condition:** User messages the seller of an item.
4. **Exit Condition:** The seller and the user agree on a price and move on to purchasing or the process is canceled.
5. **Flow of Events:**
   5.1. User messages the seller for price negotiation.
   5.2. Both parties agree on a price.
   5.3. The agreed price is confirmed by both parties during the payment process.
6. **Special/Quality Requirements:** The item must be listed as "Negotiable".

Lost & Found Items Package

1. **Name:** Post Lost Item Entry
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** The participating actor clicks the "Post Found Item Entry" button on the "Lost & Found" page.
4. **Exit Condition:** The participating actor posts a new entry or clicks cancel to close the form.
5. **Flow of Events:**
   5.1. The participating actor clicks the "Post Found Item Entry" button.
   5.2. The participating actor uploads the photo of the item if available.
   5.3. The participating actor fills in the necessary information regarding their knowledge.
   5.4. The participating actor clicks the submit button to post the new entry.

1. **Name:** Post Found Item Entry
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** The participating actor clicks the "Post Lost Item Entry" button on the "Lost & Found" page.
4. **Exit Condition:** The participating actor posts a new entry or clicks cancel to close the form.
5. **Flow of Events:**
   5.1.The participating actor clicks the "Create New Entry" button.
   5.2. The participating actor uploads the photo of the item they have found.
   5.3. The participating actor fills in the necessary information about the item found.
   5.4. The participating actor clicks the submit button to post the new entry.

1. **Name:** Delete Lost Items Entry
2. **Participating Actors: C**ampus Connect User

3. **Entry Condition:** The participating actor is the author of the entry and clicks the delete button.
4. **Exit Condition:** The entry is deleted from the system.
5. **Flow of Events:**
   5.1 The participating actor clicks the "Delete" button.
   5.2 The program brings out an alert "Are You Sure You Want To Delete?"
   5.3. The participating actor clicks "Yes".

1. **Name:** Delete Found Items Entry
2. **Participating Actors:** Campus Connect User
3. **Entry Condition:** The participating actor is the author of the entry and clicks the delete button.
4. **Exit Condition:** The entry is deleted from the system.
5. **Flow of Events:**
   5.1 The author of the entry clicks the "Delete" button.
   5.2 The program brings out an alert "Are You Sure You Want To Delete?"
   5.3. The participating actor clicks "Yes".

1. **Name:** Situation Resolved
2. **Participating Actors:** Campus Connect User
3. **Entry Condition:** The CampusConnect User is the author of the entry and clicks Situation Resolved for the related entry.
4. **Exit Condition:** The author exits the Situation Resolved page.
5. **Flow of Events:**
   5.1. The author of the entry clicks "Situation Resolved" for the related entry.
   5.2. If the entry is for a lost item,
      5.2.1. The author writes the ID of the user who found the item.
   5.3. If the entry is for a found item,
      5.3.1 The author writes the ID of the user who received the item.
   5.4. The author clicks Situation Resolved.

.

1. **Name:** Filter
2. **Participating Actors:** CampusConnect User
3. **Entry Conditions:**User is in the Lost/Found section
4. **Exit Conditions:** User is shown a list of items according to the filters applied.
5. **Flow of Events:**
   5.1. User clicks on the Filter icon.
   5.2. User selects the subject of filtration (Filter by Age) or (Filter by Type)
   5.3. User presses apply.
   5.4. The system displays a list of relevant lost/found items with their details.

1. **Name:** Browse Lost or Found Item
2. **Participating Actors:** CampusConnect Users

3. **Entry Conditions:** User is in the Lost/Found section.
4. **Exit Conditions:** User is shown the items related to the search query.
5. **Flow of Events:**
   5.1. User clicks on "Browse Item"
   5.2. The system displays a list of relevant lost/found items with their details.

Donations Package

1. **Name:** Add a Donation Request
2. **Participating Actors:** Society/Club Representative
3. **Entry Conditions:** Society/Club has a valid profile.
4. **Exit Conditions:** The request is posted.
5. **Flow of Events:**
   5.1. The participating actor writes the headline for the request.
   5.2. The participating actor shares the related information.
   5.3. The participating actor uploads the request.
   5.4. The system verifies the request and makes it visible to other users.
6. **Special/Quality Requirements:** Society/Club already has more than 3 donation requests or has a request with the same name or discards the process or adds the request successfully.

1. **Name:** Delete a Donation Request
2. **Participating Actors:** Society/Club Representative
3. **Entry Conditions:** Society/Club has an existing request.
4. **Exit Conditions:** The participating actor has deleted the request.
5. **Flow of Events:**
   5.1. The participating actor goes into the Society/Club profile.
   5.2. The participating actor selects the request that will be deleted.
   5.3. The participating actor clicks the "Delete Request" button.

1. **Name:** Edit a Request
2. **Participating Actors:** Society/Club Representative
3. **Entry Conditions:** Society/Club has an existing request.
4. **Exit Conditions:** Representative saves the new version of the request.
5. **Flow of Events:**
   5.1 The participating actor goes into the Society/Club profile.
   5.2 The participating actor selects the request to be edited.
   5.3 The participating actor makes the necessary changes on the request.
   5.4 The participating actor clicks the "Save Changes" button.

1. **Name:** View My Society/Club Profile
2. **Participating actors:** Society/Club Representative
3. **Entry Conditions:** The participating actor is logged in.
4. **Exit Conditions:** The participating actor successfully views their profile page.
5. **Flow of Events:**

5.1. The participating actor clicks on the “My Profile” option.
5.2. The system redirects the participating actor to their profile.

1. **Name:** View All Donation Requests
2. **Participating Actors:** CampusConnect Users
3. **Entry Conditions:** The participating actor is logged in.
4. **Exit Conditions:** The participating actor successfully views their donations.
5. **Flow Of Events:**
   5.1 The participating actor clicks on the donations page**.**

1. **Name:** View a Club’s Profile
2. **Participating Actors:** CampusConnect Users
3. **Entry Conditions:** Society/Club profile exists and the participating actor is logged in.
4. **Exit Conditions:** The participating actor successfully views the club’s profile.
5. **Flow of Events:**
   5.1. The participating actor searches for a Society/Club or clicks on its name on a donation request.
   5.2. The participating actor is directed to the Society/Club profile.

1. **Name:** Browse a Society/Club
2. **Participating Actors:** CampusConnect Users
3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User is shown the items related to the search query.
5. **Flow of Events:**
   5.1. User types in the club that they are interested in in the search bar.
   5.2. User clicks the “Search” button.

Authentication Package

1. **Name:** Login
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** User must be registered in the system.
4. **Exit Condition:** User successfully logs in to the system or the process is aborted.
5. **Flow of Events:**
   5.1. The user goes to the app link on their browser or is directed to it by STARS.
   5.2. The user puts in their email and password.
   5.3. If the user enters an incorrect password
      5.3.1. “Incorrect Password” message is displayed.
   5.4 Else if the user email is not valid
      5.4.1 “Invalid Email” message is displayed.
   5.4. Else the user enters the correct email and password combination and logs in successfully.

1. **Name:** Sign Up
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** User clicks the “Sign Up” button.
4. **Exit Condition:** User is successfully signed up to the system or the process is aborted.
5. **Flow of Events:**
   5.1. User enters their Bilkent University email and Bilkent ID.
   5.2. If ID or email is invalid
      5.2.1. “Invalid ID or Email” message is displayed.
   5.3. Else the user is prompted to create a password.
   5.4. User is asked to re-enter the password.
   5.5. If the two entries don’t match
      5.5.1. “Password not matched” message is displayed.
   5.6. Else the user is signed up successfully and a verification email is sent to them.
6. **Special/Quality Requirements:** System must verify that the Bilkent ID and university email are valid.

1. **Name:** Change Password
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** User clicks “Change Password” on their profile page.
4. **Exit Condition:** User successfully changes their password or process is canceled.
5. **Flow of Events:**
   5.1. An email is sent to the user with a link to the password change page.
   5.2. User is directed to the page through the link.
   5.3. User enters the new password.
   5.4. User re-enters the password to verify.
   5.5. If the two entries don’t match
      5.5.1. “Password not matched” message is displayed.
   5.6. Else user successfully changes their password.

1. **Name:** Forgot Password
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** User clicks “Forgot Password” on login page.
4. **Exit Condition**: User successfully resets their password or process is canceled.
5. **Flow of Events:**
   5.1. User enters their email.
   5.2. An email is sent to the user.
   5.3. User clicks the link in the email and is directed to the password change page.
   5.4. User enters the new password.
   5.5. User re-enters the password to verify.
   5.6. If the two entries don’t match
      5.6.1. “Password not matched” message is displayed.
   5.7. Else user successfully changes their password.

7. **Special/Quality Requirements:** The user must have a registered account in the system and the email entered for password reset must be associated with a valid, existing user account.

Profile Package

1. **Name:** View My Profile
2. **Participating Actors:** CampusConnect User
3. **Entry Conditions:** The participating actor is logged in.
4. **Exit Conditions:** The participating actor successfully views their profile page or ends the session.
5. **Flow of Events:**
   5.1. The participating actor clicks on the “My Profile” option.
   5.2. The system redirects user to their profile.
   5.3. The participating actor views and manages their profile page.

1. **Name:** View Others’ Profile
2. **Participating Actors:** CampusConnect User
3. **Entry Condition:** User is logged in.
4. **Exit Condition:** User views the profile or process is canceled.
5. **Flow of Events:**
   5.1. User searches the name on the search bar or clicks on the profile icon when viewing one of the other user’s listings.
   5.2. User is directed to the profile where they can view the name and other available information of the other user.

1. **Name:** Report User or Content
2. **Participating actors:** CampusConnect User
3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User successfully submits a report against the targeted user or content or process is canceled.
5. **Flow of Events:**
   5.1. User navigates to the targeted user's profile or content.
   5.2. User clicks on the "Report" button.
   5.3. The system prompts the user to provide details regarding the report.
   5.4. User submits the report by clicking the button “Submit”.

1. **Name:** Edit Profile
2. **Participating Actors:** CampusConnect Users
3. **Entry Condition:** User clicks the “Edit Profile” button on their profile page.
4. **Exit condition:** User clicks the “Save” button or “Cancel” button on the editing page.

5. **Flow of Events:**
   5.1. User enters their profile and clicks the “Edit Profile” button.
   5.2. All profile features are displayed with appropriate input spaces next to them (Contact Information, Biography, Profile Picture, Department, Position, etc.).
   5.3. User adds, edits or deletes any information they want.
   5.4. User clicks “Save” to save or “Cancel” to discard changes and return to the profile page.

1. **Name:** Manage Messages
2. **Participating Actors:** CampusConnect User
3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User successfully manages their messages or ends the session.
5. **Flow of Events:**
   5.1. User clicks on the “Messages” option.
   5.2. The system redirects user to the messages interface.
   5.3. User views and manages their messages.
      5.3.1. User can view received messages.
      5.3.2. User can send new messages.
      5.3.3. User can respond to messages.
      5.3.4. User can delete messages.

1. **Name:** Access Support and Contact
2. **Participating actors:** CampusConnect User
3. **Entry Conditions:** User is logged in.
4. **Exit Conditions:** User successfully accesses the support and contact or ends the session.
5. **Flow of Events:**
   5.1. User clicks on the “Support & Contact” option.
   5.2. The system redirects user to their support and contact page.
   5.3. User views frequently asked questions(FAQ).
   5.4. User can submit a support request and inquiry.

Borrowing Package

1. **Name:** Borrow an Item
2. **Participating Actors:** CampusConnect Users
3. **Entry Condition:** User is logged in.
4. **Exit Condition:** The borrowing process is successfully completed, or the user cancels the process.
5. **Flow of Events:**
   5.1. User selects an item listed for borrowing and clicks the "Borrow Now" button.
   5.2. User clicks “Agree” on the prompt repeating the borrowing conditions and the return date of the item.
   5.3. Seller receives a notification.

5.4. Seller confirms the borrowing and agrees to the terms.
5.5. The item is marked as borrowed.

6. **Special/Quality Requirements:** The system should enforce that the item is returned by the due date by dealing with penalties such as a late fee.

1. **Name:** Ask for an Extension for a Borrowed Item
2. **Participating Actors:** CampusConnect Users
3. **Entry Condition:** User has borrowed an item.
4. **Exit Condition:** Seller accepts or rejects the extension request.
5. **Flow of Events:**
   5.1. User goes to the borrowed/rented items section in their profile.
   5.2. User clicks the “Extension Request” button.
   5.3. Seller receives a notification that the borrower is asking for a 14-day extension.
   5.4. Seller goes to their profile and either click “Accept Extension Request” or “Reject Extension Request”.
   5.5. User is notified of the result.
      5.5.1. If the request is accepted the due date counter in both the user’s and the seller’s profile is updated.
6. **Special/Quality Requirements:** The system must enforce the return of the borrowed item by imposing a late fee.

Renting Package

1. **Name:** Rent an Item
2. **Participating Actors:** CampusConnect Users
3. **Entry Condition:** User is logged in.
4. **Exit Condition:** The renting process is successfully completed, or the user cancels the process.
5. **Flow of Events:**
   5.1. User selects an item and clicks the "Rent Now" button.
   5.2. User clicks “Agree” on the prompt repeating the renting conditions and the return date of the item.
   5.3. User confirms the purchase, selects the payment method, and provides necessary payment details.
   5.4. Seller receives a notification of the purchase.
   5.5. Seller confirms the sale and agrees to the terms.
   5.6. Payment is processed and the item is marked as rented.
6. **Special/Quality Requirements:** The system should ensure secure payment processing and order confirmation. The system should enforce that the item is returned by the due date by dealing out penalties such as a late fee.

1. **Name:** Ask for an Extension for a Rented Item
2. **Participating Actors:** CampusConnect User

3. **Entry Condition:** User has rented an item
4. **Exit Condition:** Seller accepts or rejects the extension request.
5. **Flow of Events:**
   5.1. User goes to the borrowed/rented items section in their profile.
   5.2. User clicks the "Extension Request" button.
   5.3. Seller receives a notification that the renter is asking for a 14-day extension.
   5.4. Seller goes to their profile and either click "Accept Extension Request" or "Reject Extension Request".
   5.5. User is notified of the result.
      5.5.1. If the request is accepted the due date counter in both the user's and the seller's profile is updated.
6. **Special/Quality Requirements:** The system must enforce the return of the rented item by imposing a late fee.

1. **Name:** Negotiate the Price of an Item for Rent
2. **Participating Actors:** CampusConnect Users, Registered Sellers
3. **Entry Condition:** User messages the seller of an item.
4. **Exit Condition:** The seller and the user agree on a price and move on to purchasing or the process is canceled.
5. **Flow of Events:**
   5.1. User messages the seller for price negotiation.
   5.2. Both parties agree on a price.
   5.3. The agreed price is confirmed by both parties during the payment process.
6. **Special/Quality Requirements:** The item must be listed as "Negotiable".

Management Package

1. **Name:** Delete User from the System
2. **Participating Actors:** Admin
3. **Entry Condition:** Admin enters the management panel.
4. **Exit Condition:** Admin deletes CampusConnect User.
5. **Flow of Events:**
   5.1. Admin chooses a user to delete from the system.
   5.2. From the opened page, admin deletes the desired user from the system by clicking the delete button.

1. **Name:** Change Time Zone
2. **Participating Actor:** Admin
3. **Entry Condition:** Admin enters the management panel.
4. **Exit Condition:** Admin exits the management panel.
5. **Flow of Events:**
   5.1. Admin decides to change the time zone of the system.
   5.2. Admin picks the new time zone and clicks the save button.

1. **Name:** Change Language
2. **Participating Actor:** Admin
3. **Entry Condition:** Admin enters the management panel.
4. **Exit Condition:** Admin exits the management panel.
5. **Flow of Events:**
   5.1. Admin decides to change the language of the system.
   5.2. Admin picks the new language and clicks the save button.

1. **Name:** View System Information
2. **Participating Actor:** Admin
3. **Entry Condition:** Admin enters the management panel.
4. **Exit Condition:** Admin exits the management panel.
5. **Flow of Events:**
   5.1. Admin decides to view the system information of the system.
   5.2. Admin clicks the view system information button and sees the details.

TECH STACK

- Front End: JavaScript/React

  React is a popular and highly efficient JavaScript library for building user interfaces. It's chosen for its component-based architecture and ease of development, providing a responsive and interactive user experience.

- Back End: SpringBoot / Java

  Spring Boot, built on the Java platform, is known for its simplicity and convention-over-configuration approach, allowing developers to quickly build robust and scalable web applications. Java is renowned for its portability, strong type system, and extensive libraries, making it a dependable choice for server-side development. Spring Boot's features, like auto-configuration and embedded web servers, streamline development and deployment processes, which is essential for creating efficient back-end systems.

- Database: MySQL

  MySQL is a well-established, open-source relational database management system known for its reliability, performance, and ease of use. It's chosen for data storage and retrieval due to its strong community support and compatibility with a wide range of applications.

NONFUNCTIONAL REQUIREMENTS

User Interface and Human Factors:

CampusConnect is designed for a diverse user base within the university community, including students, faculty members, and staff, meaning multiple user types will utilize the system. The system's interface should be intuitive and easy to learn, as it aims to enhance the user experience so it wouldn't require any training to use the website. The color theme would be similar to Bilkent University's WEB page and SRS page to create familiarity for the users. The titles and subtitles would be bold and scaled to fit the context and capture the users' attention. Instruction will be short and clear to avoid any kind of misunderstanding. Short does not always mean it is clear so the descriptions will be detailed enough. The descriptions can be supported with bullet points to make them easier to follow. Users would be protected from making errors by guiding them through the purchase process and sending several notifications on significant actions to provide a safe feedback system. The system supports a variety of input/output devices, including web browsers on computers and mobile devices, to accommodate different user preferences and accessibility needs.

Documentation:

Comprehensive documentation is essential to support users and administrators. For end-users, which includes students, faculty, and staff, the documentation should be user-centric, offering clear and comprehensive guidance on how to use the platform effectively. This documentation may include user guides, FAQs, and video tutorials that supply varying learning preferences. Administrators would also be provided with the necessary documentation to ensure the system's smooth operation and to address issues that may arise. Administrative documentation would encapsulate instructions on system configuration, user management, reporting tools, system monitoring, and the best practices for ensuring data security and system integrity, as administrators play a critical role in the trust and reliability of CampusConnect. Developer documentation would also most possibly be needed for the technical team tasked with ongoing system development and maintenance. It would include details about the system architecture, coding standards, APIs, and other technical aspects; making it easier to implement enhancements, bug fixes, and updates.

Hardware Considerations:

CampusConnect would be designed to operate on standard hardware, including personal computers, laptops, and mobile devices. Hardware characteristics such as memory size and auxiliary storage space must be considered to ensure optimal performance and scalability on various devices to accommodate the user base.

Performance Characteristics:

Ensuring the responsiveness and scalability of the CampusConnect system is fundamental to meeting the diverse and dynamic needs of its university community users. The system must be

capable of efficiently handling at least 1000 concurrent users, as universities often experience peak usage periods, such as the start of a new semester or during events. When the system has reached the limit of 1000 active users, the users trying to log in should be taken into a queue to sustain the performance of the already logged-in users. As already logged-in users log out from the system, users in the queue should be allowed to the system based on their order in the queue. In order to ensure an enjoyable and frustration-free user experience, responsiveness entails not just low-latency responses to user inputs but also constant and predictable performance. Response time would definitely be optimized to make interactions feel instantaneous and throughput would be robust enough to handle peaks in user activity without delapsing in performance. CampusConnect would also need to efficiently store and retrieve data associated with classified listings, lost and found items, user profiles, and much more. This data may grow over time as the user base expands and the system accumulates information. Thus, an effective data management strategy, encompassing efficient database design is crucial to maintaining system performance and ensuring that users can readily access the information they need.

Error Handling and Extreme Conditions:

For everyday usability, clear and informative error messages play a crucial role. When users make mistakes or encounter issues, the system's error messages provide precise guidance, aiding users in understanding the problem and taking the right actions. Additionally, the system incorporates data validation techniques to proactively prevent erroneous data entry. For instance, after a user lists an item, the listing is first checked by the system to prevent any invalid or incorrect data. By checking and validating user inputs, the risk of invalid or harmful data compromising the system's integrity is significantly reduced. In extreme conditions, such as system failures or surges in traffic, CampusConnect adopts a proactive approach. It employs fallback mechanisms to ensure that critical functions remain operational even in adverse situations. Moreover, the system is designed to smoothly decrease, which means that it can adjust to adverse situations by minimizing the impact on users while maintaining critical functionality. The system can efficiently recover without losing data in the case of a system breakdown or high traffic volume, protecting the integrity of user transactions and data.

System Interfacing:

The system is designed with the flexibility and adaptability to interface with a variety of external systems that serve different purposes, from payment processing to communication services. One key area of external system integration is with payment gateways. To facilitate secure transactions for users buying and selling items within the platform, CampusConnect integrates with reputable payment processors. Compliance with these standards helps safeguard sensitive user information, such as credit card details, fostering trust and security. Another critical aspect of integration involves messaging services. CampusConnect incorporates messaging functionalities to enable communication between users for item inquiries, negotiations, and other interactions. Furthermore, the system is designed to comply with any format or medium restrictions imposed by external systems for both input and output data. This means that data

exchange with these systems occurs in a structured and standardized manner, preventing compatibility issues and data loss.

Quality Issues:

CampusConnect offers a local network to everyone within the Bilkent University community. The system must prioritize reliability to maintain uninterrupted service for users. In order the drive the people to actively use this application for their needs, it needs to uphold a standard of quality and credibility. As such it will be capable of trapping faults, with minimal downtime in the event of a failure, and meet defined service-level agreements regarding system uptime. As CampusConnect can be accessed at any time by any user it is imperative that the system stays up consistently. Thus we will be aiming for a minimum 95% uptime, which should be tracked to improve performance. The CampusConnect system is portable since it is a web application and can be reached from any device with an internet connection and a browser.

System Modifications:

Parts of the system are likely to undergo modifications as the platform evolves to meet the dynamic needs of the university community. These modifications can include feature enhancements, bug fixes, and security updates. Since the system will be designed in an object-oriented manner, later modifications to the system are not expected to disrupt user experience. The modular structure and the use of concepts such as polymorphism will facilitate these processes, especially bug fixes. Some features most likely to require future modifications are:

- Categories of second-hand items according to trends in usage.
- Adding two-factor authentication to strengthen security.
- AI policing of reported users and listings.
- Advancements in payment processing by card.

Security Issues:

CampusConnect offers a variety of services to users including selling and purchasing items, viewing information about university clubs and societies, and accessing another user's profile. All of these functionalities require data privacy and security measures. Access to data and the system should be controlled to protect user information and transactions. This includes user authentication, authorization, and data encryption to ensure data integrity. While the users can shape their profile the way they want, in order to protect their privacy, only their names will be available on their profile by default. Each user will have to open their account with their Bilkent University email address and Bilkent ID to prevent one person from operating from multiple accounts, as well as keep track of poor behavior or foul language used through the system. The user agreement should contain a clause that allows the monitoring of the personal chats by admins in any kind of report since this site is not for personal discussions but trade between campus connect users. This clause aims to reduce the scams between users and to ensure the security of the users. System logs should be kept to monitor user activity and detect any kind of account theft. Online payments should comply with PCI DSS. This should be possible with an

already trustworthy online payment supplier. This will guarantee a safe and pleasant environment for the university community to interact. Physical security is not imperative since the system will be accessible through the Cloud.

Resources and Management Issues:

Regular system backups are crucial to protect data so the system will be backed up automatically every 24 hours. To ensure continuous support and improvement a dedicated team must maintain the system. After the project is finished installation, maintenance, and administration will be done by Bilkent University personnel. Their responsibilities will include technical support, updates, and patch management to keep the system running smoothly and handle any future problems. The administrators, with the guidance of the Administrative Document, will investigate reported users or listings within the system and make decisions on whether to report to the university. Suspicious activity through the program can be identified and penalized by the university.