CS319 – Object Oriented Software Engineering

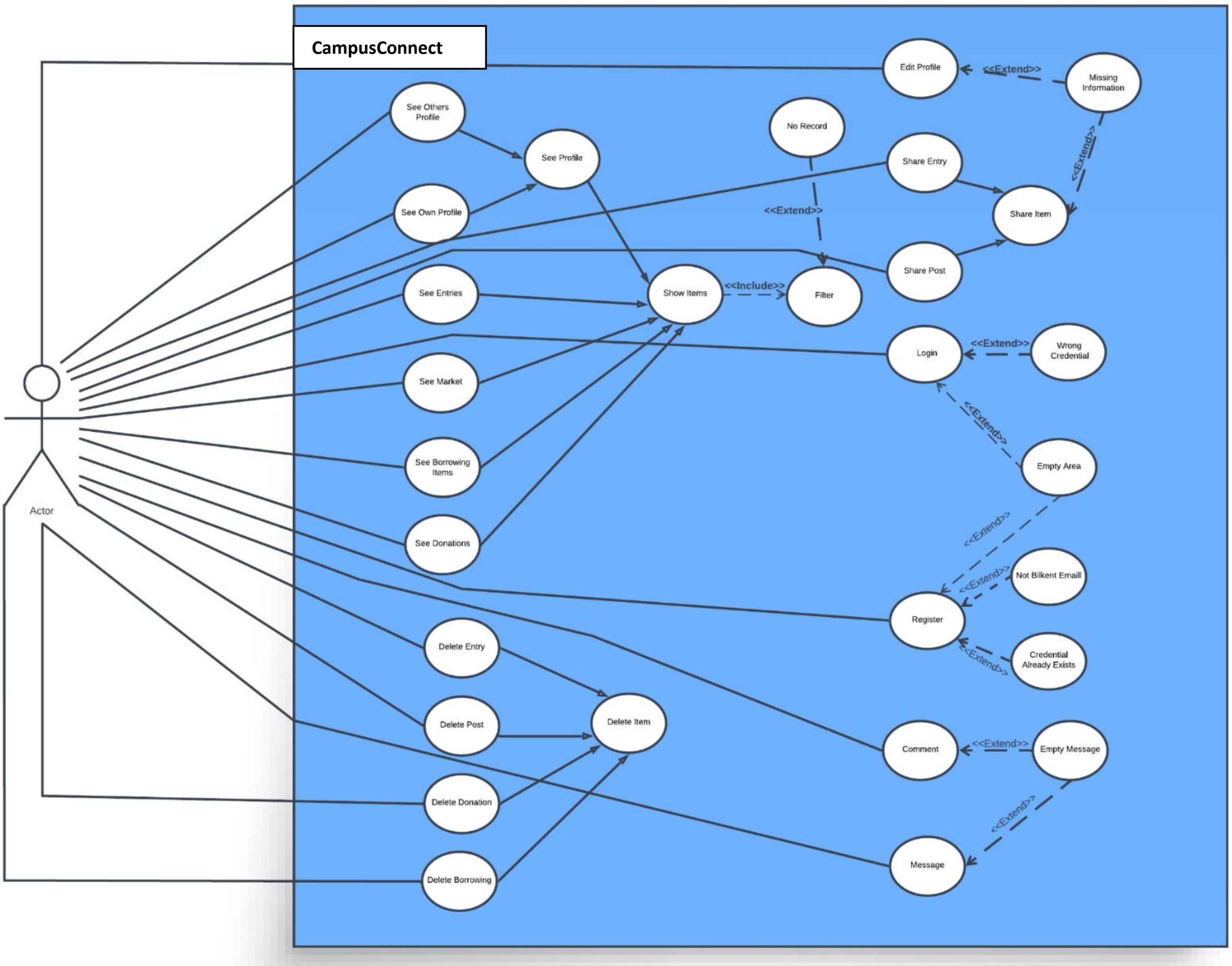
Section 3

Team 4 – FACID

Analysis Report

Çağatay **AKPINAR 22003508**
İlhami **ULUĞTÜRKKAN 22102546**
Alphan **TULUKCU 22003500**
Feza Emir **ÇELİK 22101910**
Deniz Can **ÖZDEMİR 22003854**

1.0 Use Case Diagram



Use Case Name: Login

Participating Actor: User

Flow Of Events:

- User enters username and password.
- If one of the parts is empty
  - “Please fill this area” message pops up and the process continues until the user fills it.
- If the username or the password is not found in database
  - “Check your username and password.” message occurs and the login process restarts.
- If both username and password is found and correct, the user logs in successfully.

Entry Conditions: Opening the app through the internet.

Exit Conditions: User logs in or the login process fails.

Use Case Name: Register

Participating Actor: User

Flow Of Events:

- User fills the username, name, surname, mail and password parts.
- If one of the parts is empty
  - "Please fill this area" message pops up and the process continues until the user fills it.
- If the user does not use his/her bilkent mail
  - "Please Enter Your Bilkent Mail" message pops up and the registration process restarts.
- If all the information is proper
  - a verification mail goes to the relevant mail address.

Entry Conditions: User clicks the button "Register" on the "Log-In" page.

Exit Conditions: User clicks the "Back to Log In" button or register process fails.

Use Case Name: Verificate Account

Participating Actor: User, Database

Flow Of Events:

- Database system sends a verification code to the Bilkent mail of user.
- User enter this verification code to the verification page of CampusConnect
- If entered verification code is matched with the verification code in mail.
  - Account is succesfully verified
- If entered verification code is matched with the verification code in mail.
  - Account is not succesfully verified and error pop-up appears in screen.

Entry Conditions: User clicks the button "Verificate Account" on the "Main" page.

Exit Conditions: User clicks the "Enter code" button correctly or enter wrong code.

Use Case Name: See Own Profile

Participating Actor: Actor

Flow Of Events:

- If a user wants to see its accounts properties, he/she clicks the “Profile” button on the menu.
- All of his/her shared posts and entries show up on the page.
  - If the user does not have any shared entries or posts
    - “No entries” and “No posts” messages occur.

Entry Conditions: User clicks “Profile” button on the menu.

Exit Conditions: User clicks another page's button.

Use Case Name: See Others’ Profile

Participating Actor: User

Flow Of Events:

- User sees wanted user’s shared items and account properties.

Entry Conditions: User clicks to another user’s name through its shared post’s or entry’s detailed page.

Exit Conditions: User clicks another page’s button.

Use Case Name: Edit Profile

Participating Actor: User

Flow Of Events:

- User edits his/her profile’s properties.
- If a property is not filled
  - “Please fill this area” message pops up and the editing process continues.
- If changed username already exists
  - “Username already exists” message pops up and the editing process continues.
- If all the parts filled
  - “Please fill this area” message pops up and the editing process continues.

Entry Conditions: User clicks the “Edit Profile” button on the “Profile” page.

Exit Conditions:

1. If the editing process ends successfully and the user returns to the “Profile” page.
2. If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Add Entry

Participating Actor: User

Flow Of Events:

- Click “Add” button on main page.
- Select entry from “Entry” button
- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture) for new entry.
- If one of the mandatory areas is empty.
  - “Please fill all mandatory areas” message pop-up and adding entry process continues.
- If all the parts filled properly
  - Entry is succesfully shared.

Entry Conditions: User clicks the “Add” button on the “Main” page.

Exit Conditions:

1.If the adding entry process ends successfully and the user returns to the “Profile” page.

2.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Add Post

Participating Actor: User

Flow Of Events:

- Click “Add” button on main page.
- Select post from “Post” button
- Fill the areas (Title, Description, Category, Subcategory, Anonymous, Picture, Price) for new entry.
- If one of the mandatory areas is empty.
  - “Please fill all mandatory areas” message pop-up and adding entry process continues.
- If all the parts filled properly
  - Post is succesfully shared.

Entry Conditions: User clicks the “Add” button on the “Main” page.

Exit Conditions:

1.If the adding post process ends successfully and the user returns to the “Profile” page.

2.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: See Entries

Participating Actor: User

Flow Of Events:

- Click “Forum” button on main page.
- View all entries from “Forum” page.
- User filter entries according to category and subcategory
- If there is entries which is proper to filters
  - User view entries according to his/her filters
- If there is no entries which is proper to filters
  - User view empty forum page.

Entry Conditions: User clicks the “Forum” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: See Market

Participating Actor: User

Flow Of Events:

- Click “Market” button on main page.
- View all posts from “Market” page.
- User filter posts according to category and subcategory
- If there is posts which is proper to filters
  - User view posts according to his/her filters
- If there is no posts which is proper to filters
  - User view empty market page.

Entry Conditions: User clicks the “Market” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: See Borrowing Items

Participating Actor: User

Flow Of Events:

- Click “Borrow” button on main page.
- Click “All Borrowings” button from “Borrow” page.
- View all borrowing items from “All Borrowings” page.
- User filter borrowings according to category and subcategory
- If there is borrowing which is proper to filters
    - User view borrowings according to his/her filters
- If there is no borrowing which is proper to filters
    - User view empty “All Borrowings” page.

Entry Conditions: User clicks the “Borrow” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: See Donations

Participating Actor: User

Flow Of Events:

- Click “Donation” button on main page.
- Click “All Donations” button from “Donation” page.
- View all donations from “All Donations” page.
- User filter donations according to category and subcategory
- If there is donations which is proper to filters
    - User view donations according to his/her filters
- If there is no donation which is proper to filters
    - User view empty “All Donations” page.

Entry Conditions: User clicks the “Donation” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Make Comment

Participating Actor: User

Flow Of Events:

- Click “Forum” button on main page.
- View all entries from “Forum” page.

- Clicks one of the entries from “Forum” page.
- Write comment to the comment area below the entry.
- Click “Comment” button and send comment.

Entry Conditions: User clicks the “Forum” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

2.If the user comment succesfully and directed “Forum” page again.

Use Case Name: Send Message

Participating Actor: User

Flow Of Events:

- Click “Market” button on main page.
- View all posts from “Market” page.
- Clicks one of the posts from “Forum” page.
- Click “Message” button below the post.
- Directed to “Chat” page and start messaging.

Entry Conditions: User clicks the “Market” button on the “Main” page.

Exit Conditions:

1.If the user clicks to another page’s button on the menu, the user goes to the relevant page.

Use Case Name: Delete Entry

Participating Actor: User

Flow Of Events:

- Click “Profile” button on main page.
- User view his/her profile informations from “Profile” page.
- Clicks his/her entries from “Profile” page.
- Click “Delete” button on “Entry” page.
- Entry is succesfully deleted.

Entry Conditions: User clicks the “Profile” button on the “Main” page.

Exit Conditions:

1.If the user clicks to “Delete” button on the “Entry” page.

Use Case Name: Delete Post

Participating Actor: User

Flow Of Events:

- Click “Profile” button on main page.
- User view his/her profile informations from “Profile” page.
- Clicks his/her posts from “Profile” page.
- Click “Delete” button on “Post” page.
- Post is succesfully deleted.

Entry Conditions: User clicks the “Profile” button on the “Main” page.

Exit Conditions:

1.If the user clicks to “Delete” button on the “Post” page.

Use Case Name: Delete Donation

Participating Actor: User

Flow Of Events:

- Click “Profile” button on main page.
- User view his/her profile informations from “Profile” page.
- Clicks his/her donations from “Profile” page.
- Click “Delete” button on “Donation” page.
- Donation is succesfully deleted.

Entry Conditions: User clicks the “Profile” button on the “Main” page.

Exit Conditions:

1.If the user clicks to “Delete” button on the “Donation” page.

Use Case Name: Delete Borrowing

Participating Actor: User

Flow Of Events:

- Click “Profile” button on main page.
- User view his/her profile informations from “Profile” page.
- Clicks his/her borrowings from “Profile” page.
- Click “Delete” button on “Borrowing” page.
- Borrowing is succesfully deleted.

Entry Conditions: User clicks the “Profile” button on the “Main” page.

Exit Conditions:

1.If the user clicks to “Delete” button on the “Donation” page.

2.0 Non-Functional Requirements

2.1 Performance

Performance is the heartbeat of CampusConnect, with every aspect meticulously tuned to ensure lightning-fast interactions. Our login and logout processes are designed to take less than 2 seconds, allowing users to access their accounts swiftly. Navigating between pages is a seamless experience, taking less than 2 seconds to transition effortlessly. Whether you're uploading a post or entry with images, we prioritize speed – pictures from our database to your screen in under 5 seconds and post creation in less than 2 seconds, even with photographs. In the background, the entire system data is dynamically backed up to ensure data integrity and reliability. The importance of such performance measures cannot be overstated. A rapid, responsive system enhances user satisfaction, engagement, and the overall success of our platform. With CampusConnect, every second counts in making your online campus experience exceptional. These performance measures are vital to ensure that users have a smooth and efficient experience when using the CampusConnect platform. The speed of these actions directly impacts user engagement and the overall functionality of the website. Rapid performance is key to user satisfaction and success in providing an exceptional online campus experience.

2.2 Safety/Security

At CampusConnect, security and safety are paramount. We've implemented stringent measures to safeguard user data, ensuring that personal information is encrypted and stored securely within our database. In CampusConnect password of users stored in database as hashed form so even admins cannot reach the passwords of users. Additionally, the app boasts robust authentication and authorization mechanisms, limiting access exclusively to individuals within the Bilkent University community. Account verification feature of our app provide the chance of creating in-campus community in CampusConnect app. The importance of these security measures cannot be overstated. They are the foundation of user trust and confidence in our platform. As guardians of sensitive data, we prioritize the protection of personal information and access controls to ensure that CampusConnect remains a safe and secure digital space for our users, fostering trust, peace of mind, and a thriving online community.

2.3 Usability

Usability is a cornerstone of CampusConnect's design philosophy, and it holds immense importance in delivering an exceptional user experience. Our user interface is meticulously crafted to be intuitive, accessible, and responsive, catering to users of different abilities. We have focused on creating an easy-to use and simple user interface that ensures a seamless experience. Places of all buttons are on the top menu and easy to find&use for all users. Also while adding post, entry, borrowing and donation; blanks which must be filled are located at the right of page. The app's compatibility with popular web browsers, including Chrome, Safari, and Opera, guarantees that users can access CampusConnect with ease using any web browsers. Additionally, we've chosen color tones (Orange&White) that are easy on the human eye to prevent visual fatigue and ensure a pleasant reading experience. Our buttons are designed to be clear and user-friendly, streamlining interactions. By prioritizing usability, we aim to provide an accessible, enjoyable, and efficient platform for all users, promoting engagement and making CampusConnect a user's first choice.

2.4 Reliability

Reliability is at the core of CampusConnect, and its significance cannot be overstated. Our commitment to offering a dependable platform means that the

CampusConnect app will be available 24/7, with minimal downtime reserved exclusively for essential maintenance. Furthermore, we've taken measures to ensure the system can recover gracefully from any unforeseen system failures, all without the loss of any critical data. We understand that reliability is the linchpin of user trust and satisfaction. It's the assurance of uninterrupted service, data integrity, and a seamless experience for our users. By prioritizing reliability, we aim to provide a platform that users can depend on, day in and day out, fostering a strong sense of trust and loyalty within our digital community.

3.0 Tech Stack

3.1 Back-End (Server Side):

Python: All back-end stage of Campus Connect is based on Python, and its frameworks for web development. Python is a high-level programming language which is known as readable, and it supports various programming approaches such as object-oriented, functional etc. It has widely usage in web development, artificial intelligence, data analysis, and automation. One of the most significance advantages is that its extensive libraries and frameworks make backend development more efficient.

Django: Campus Connect is a web-based project arised with Django framework. Django is a high-level Python web framework providing almost everything a web-project needs such as object modeling (ORM), admin panel, and authentication support. Suitable for building scalable and maintainable web applications.

Django Rest Framework (DRF): Django Rest Framework is used for building APIs for Campus Connect. Its features include serialization for ORM and non-ORM data sources, authentication policies, and extensive documentation. Ideal for developing complex, database-driven websites and enabling web services/APIs.

3.2 Front-End (Client Side):

HTML: HTML is supported by Django framework to provide a user interface after API called. The standard markup language used to create web pages. Its main advantage for Campus Connect is to design all the web content and connect with back-end with calling urls with respect to users choice for all web pages.

CSS: CSS, namely Cascading Style Sheets is used for make the style of the HTML contents better and more specific. It can control the layout of multiple web pages all at once so that the design pattern of the Campus Connect is more stable.

JavaScript: JavaScript is used in front-end stage of Campus Connect. It makes the HTML pages be responsive and increases its functionality.

Bootstrap: A front-end framework for developing responsive and advanced user interfaces. It includes HTML and CSS design templates for various interface components, and with some JavaScript extensions. Campus Connect front-end stage contains several Bootstrap components in particular pages.

3.3 Database:

MySQL: MySQL is a relational database management system that uses structured query language (SQL) for accessing and managing the data stored in relational databases. Campus Connect is based on ORM objects such as profile, post, entry, category objects etc., which are stored in MySQLdatabase provided by Amazone Web Services.

Google Firebase (Pyrebase4 Framework): Firebase is a platform developed by Google for storing visual components, and each image in Campus Connect is stored in there. Pyrebase4 is a Python wrapper for the Firebase API, and it ensure the storing images with basic back-end code parts.

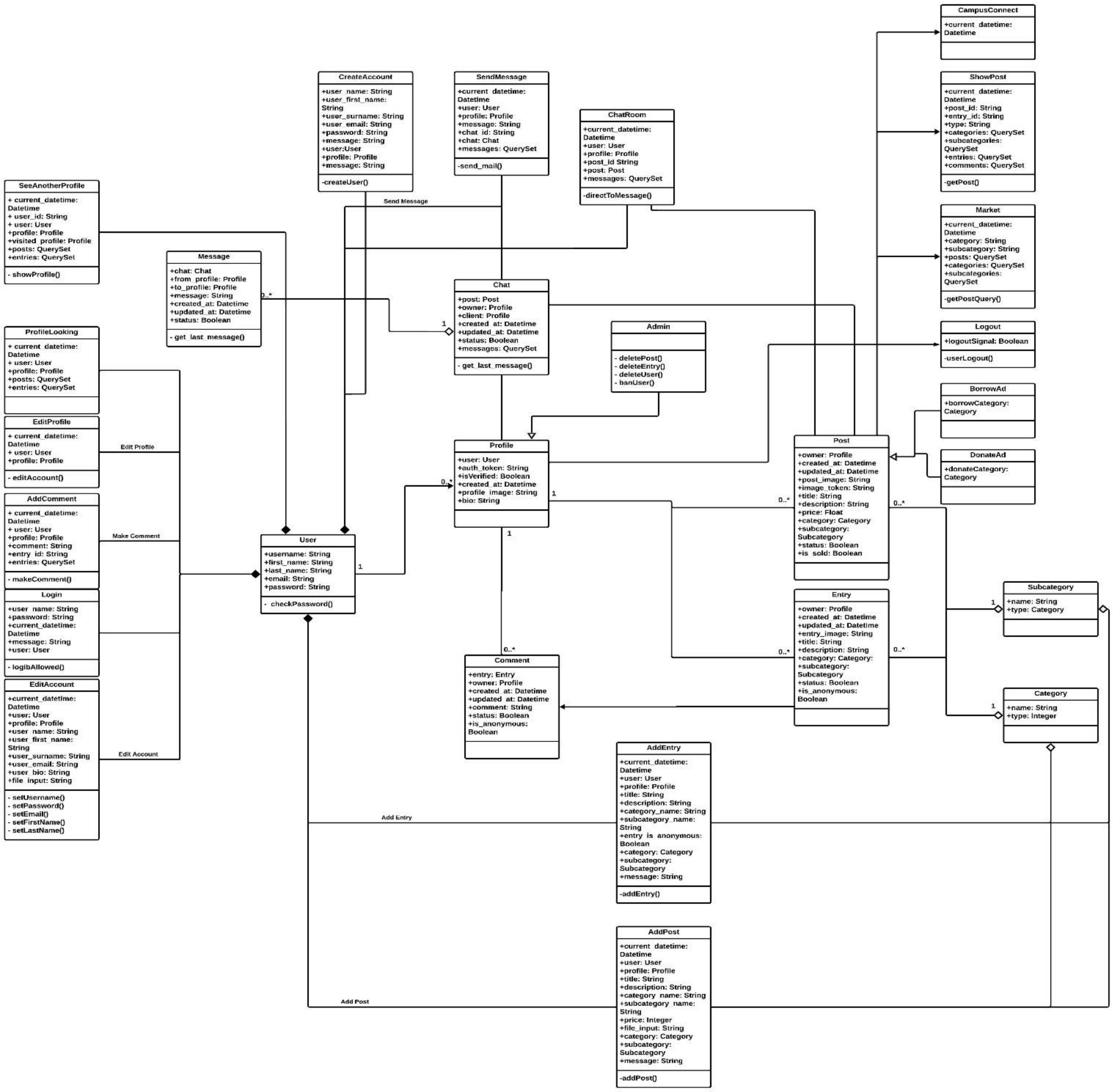
3.4 Server Infrastructure:

EC2 (Amazon Web Services): EC2 (Elastic Compute Cloud) is a part of Amazon's cloud-computing platform allowing users to use virtual machines for various applications to deploy the web-applications. Campus Connect is planned to deploy a virtual machine with EC2.

3.5 Version Control and DevOps Tools:

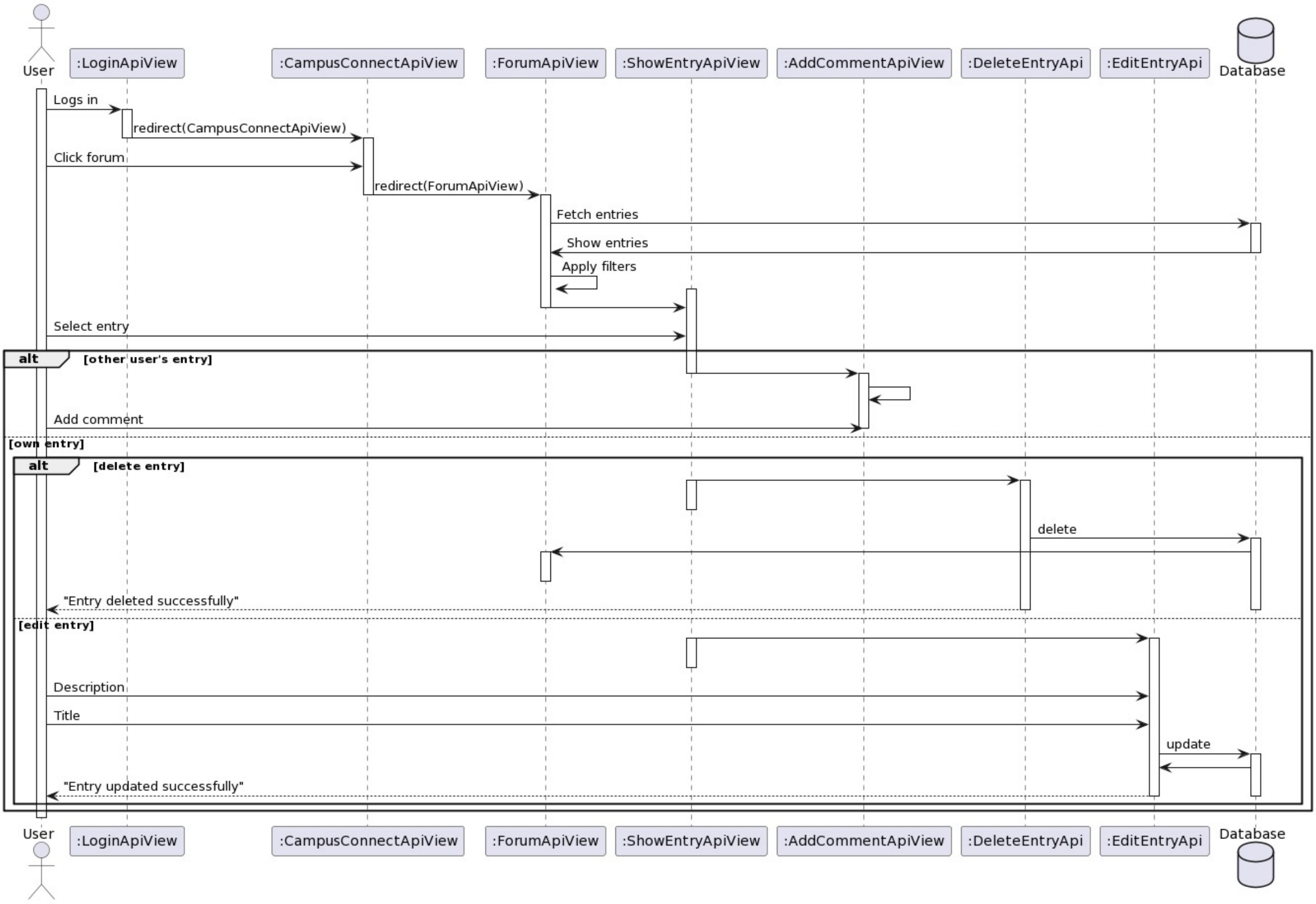
Git: A distributed version-control system for tracking changes in source code during software development. All of the version controls and code sharing is handled with Git and GitHub for Campus Connect.

4.0) Class Diagram

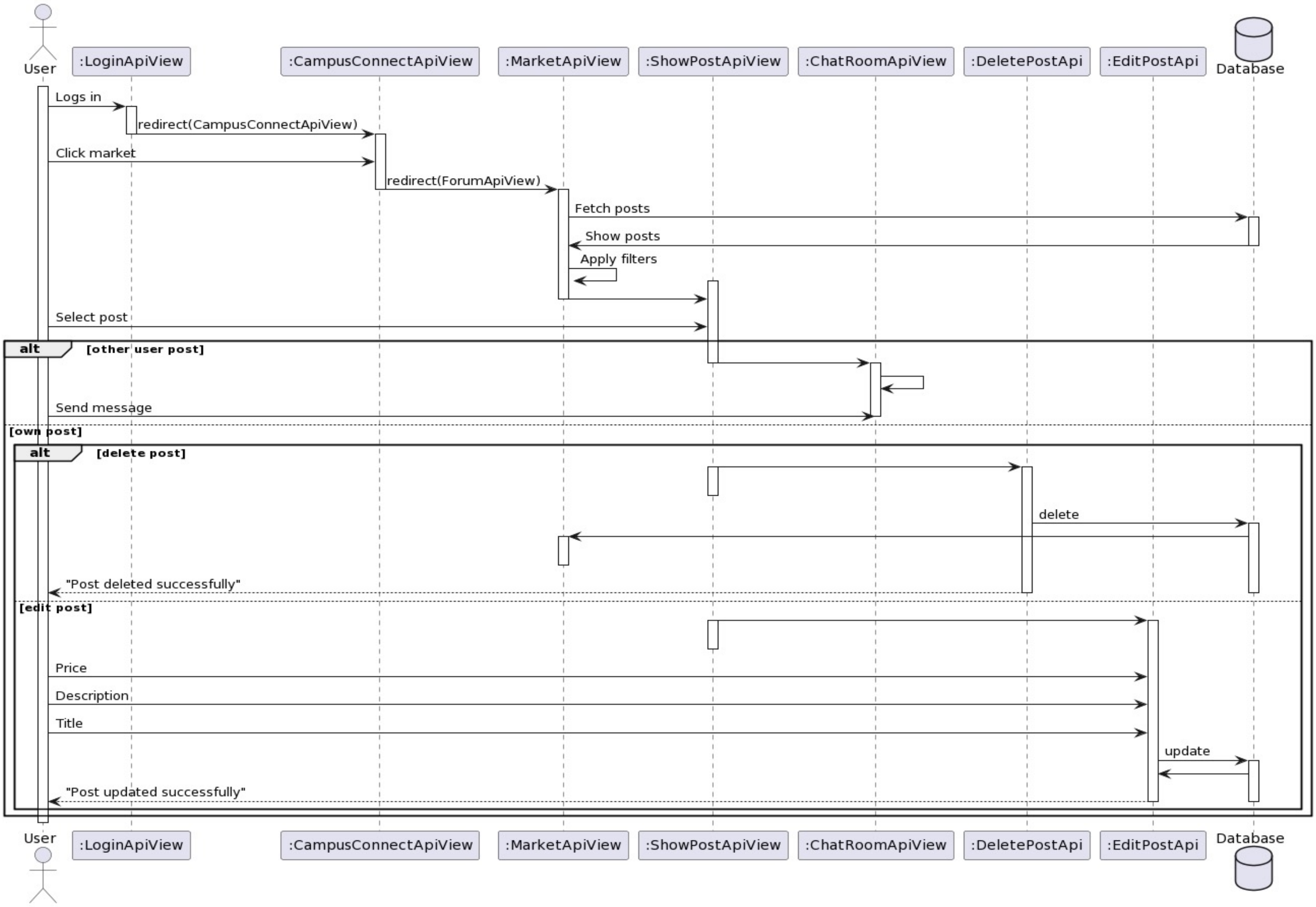


5.0)Sequence Diagrams

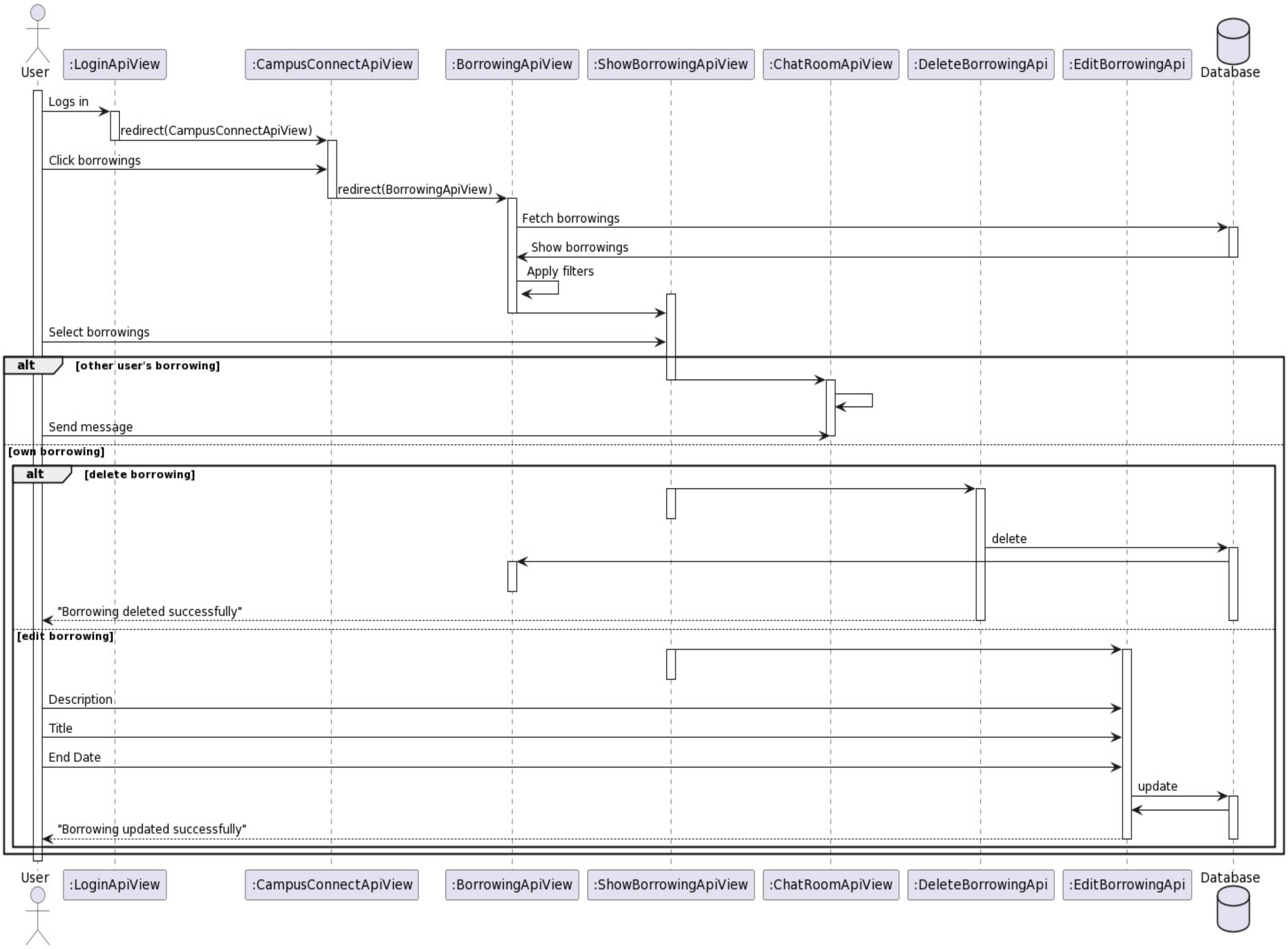
5.1) Sequence Diagram of Forum Feature



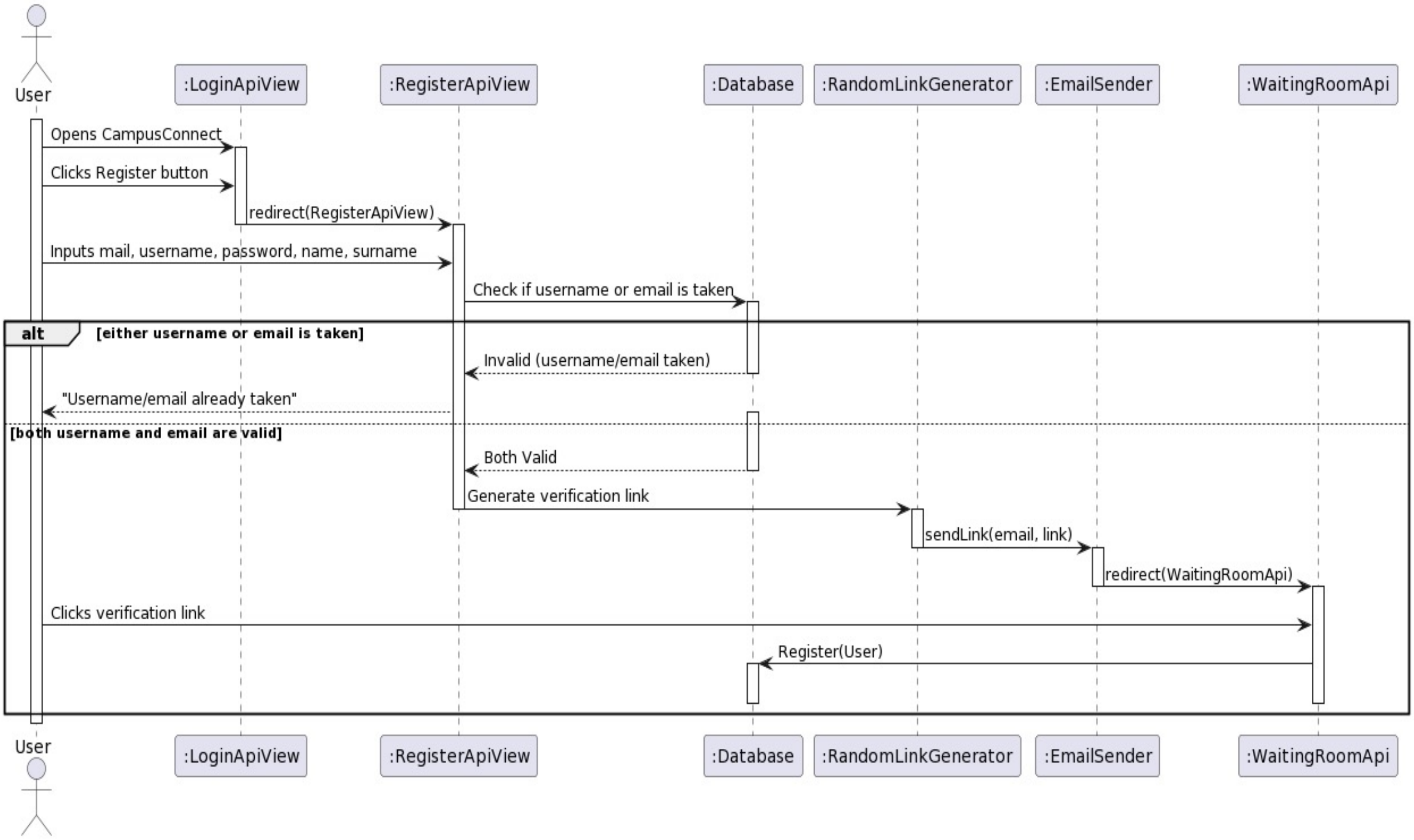
5.2) Sequence Diagram of Market, Editing and Deleting Features



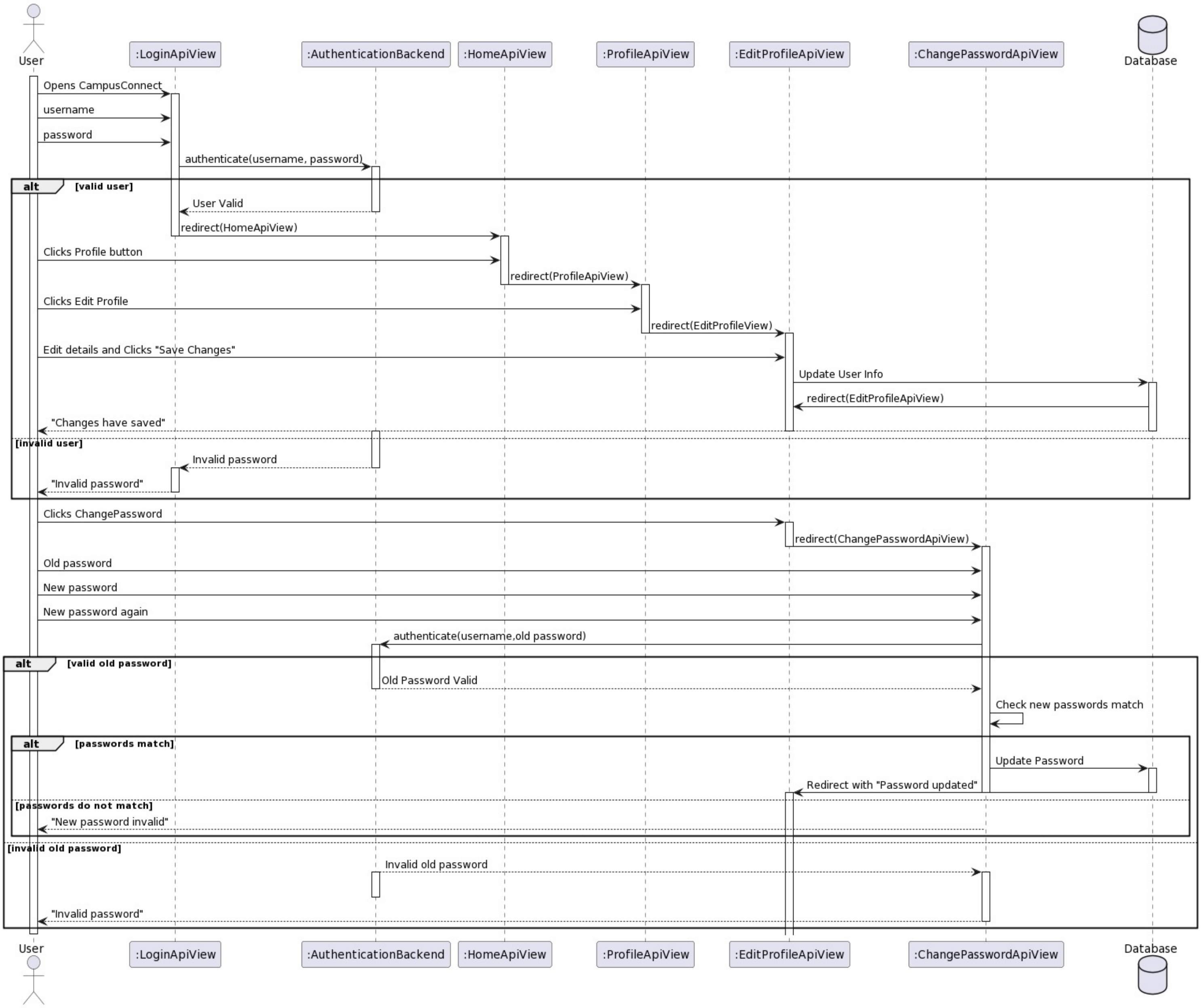
5.3) Sequence Diagram of Borrowing Features



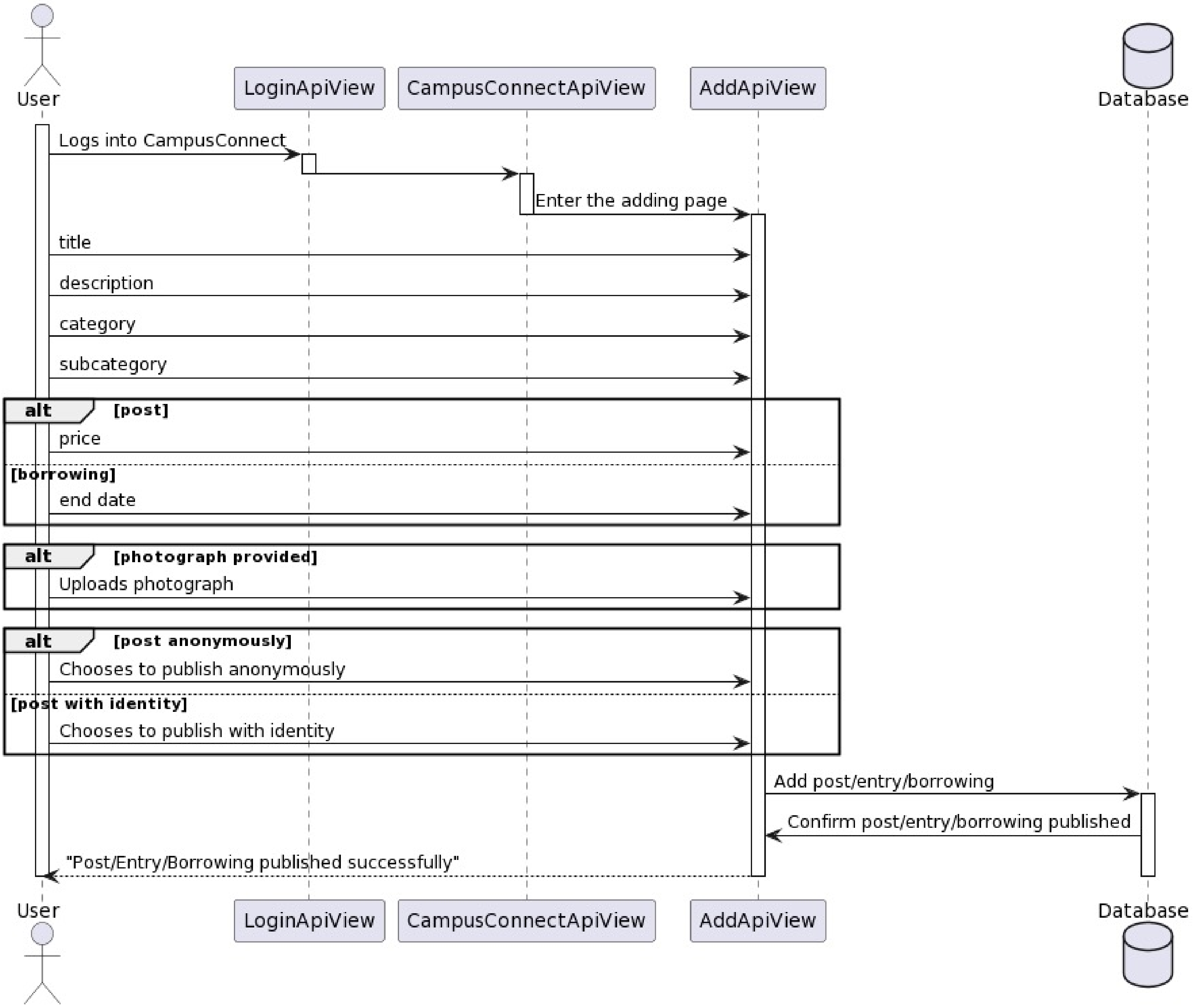
5.4) Sequence Diagram of Register Feature



5.5) Sequence Diagram of Edit Profile and Login features

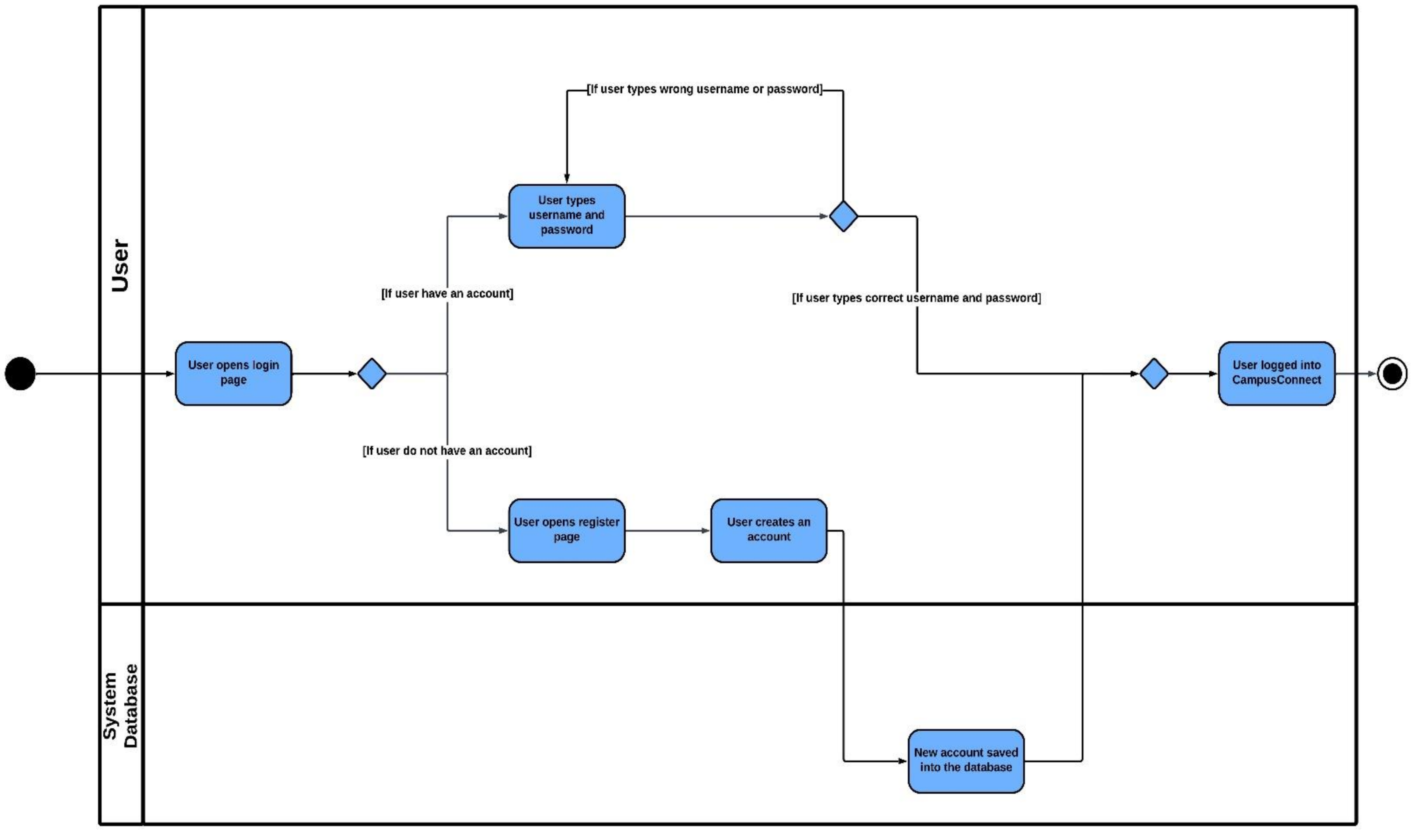


5.6) Sequence Diagram of Adding Features

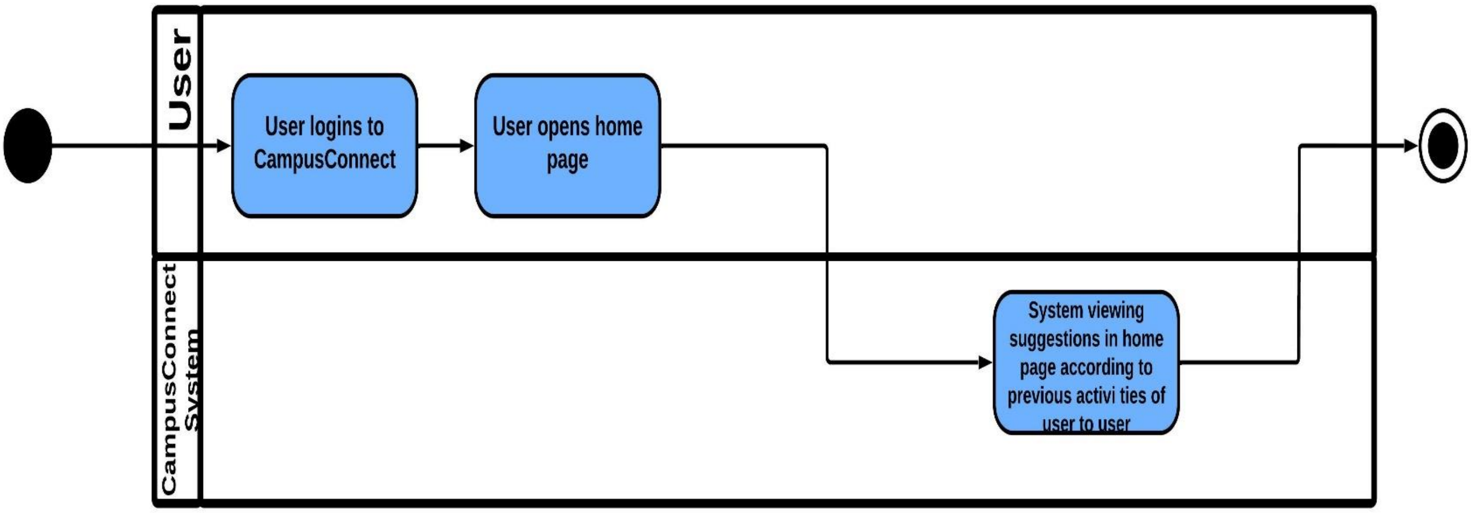


6) Activity Diagrams

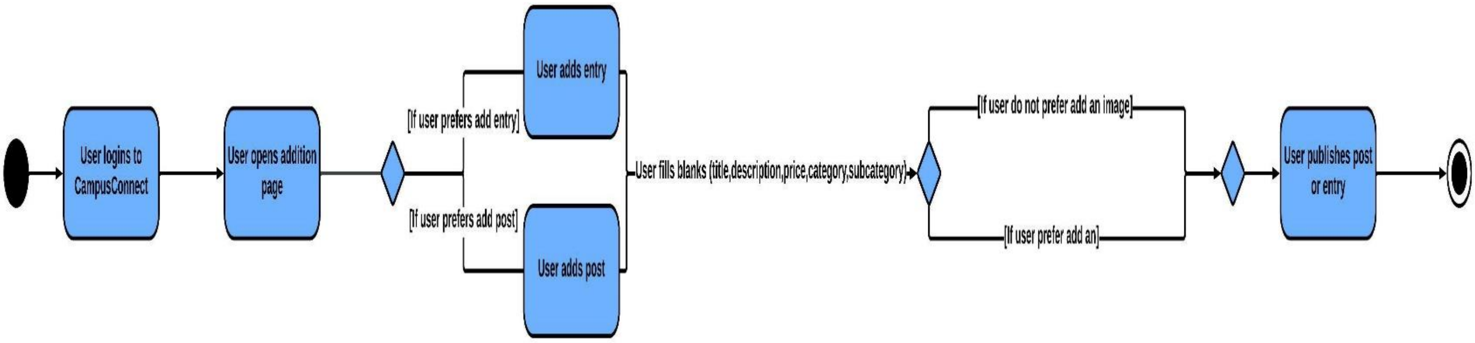
6.1) User Logins Into his/her Account


User
System Database
User opens login page
[If user have an account]
[If user do not have an account]
User types username and password
[If user types wrong username or password]
[If user types correct username and password]
User opens register page
User creates an account
New account saved into the database
User logged into CampusConnect

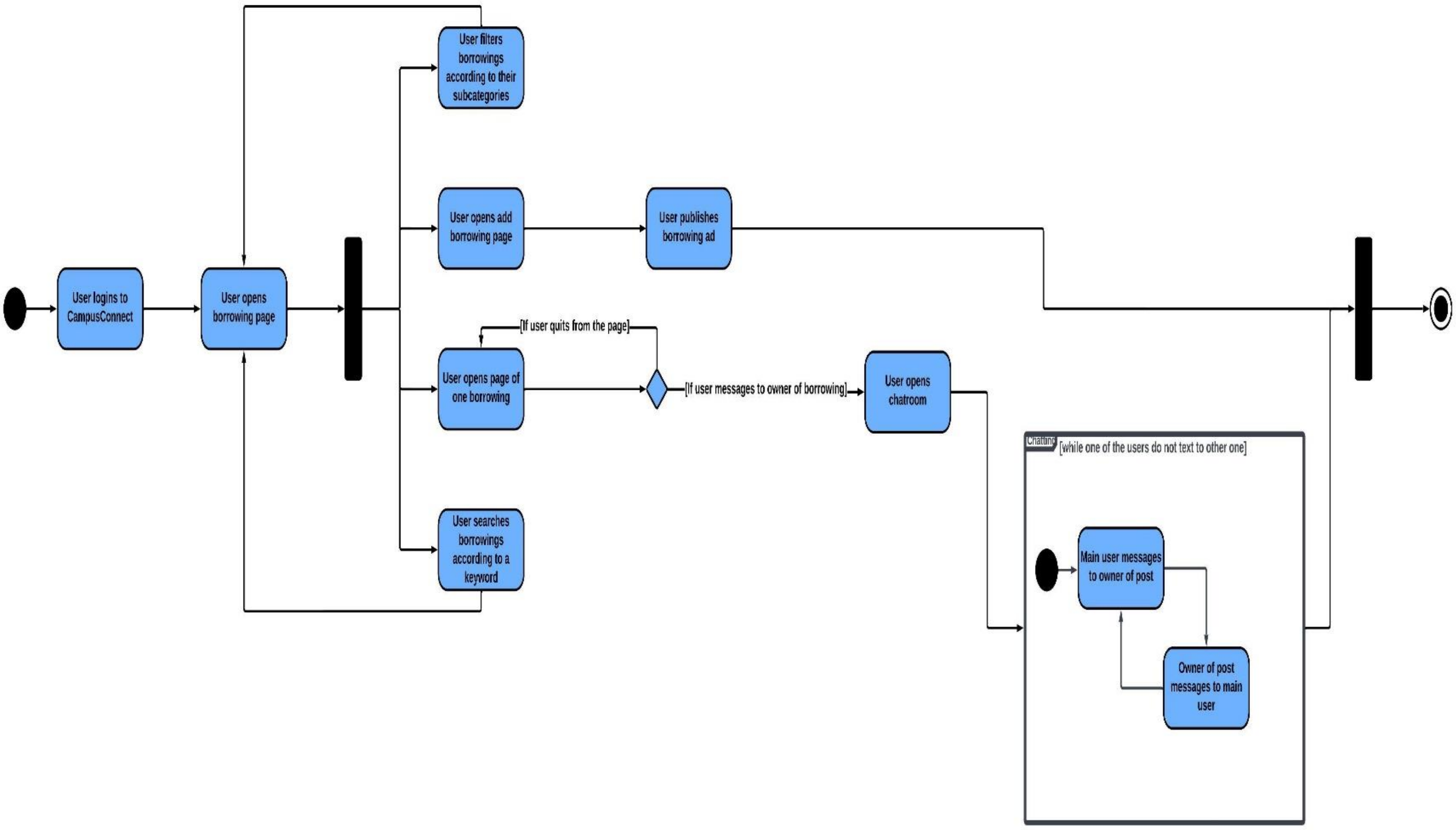
6.2) User Views Suggestions at Main Page


User
CampusConnect System
User logins to CampusConnect
User opens home page
System viewing suggestions in home page according to previous activi ties of user to user

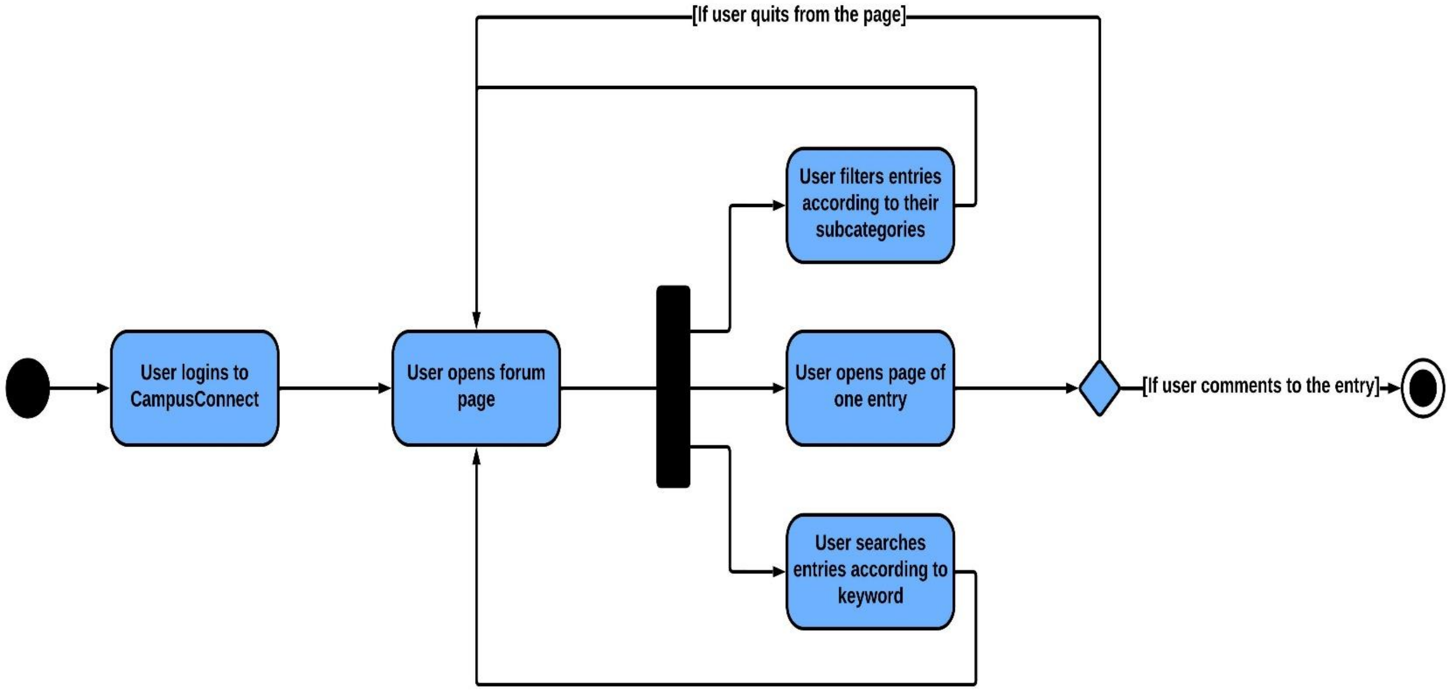
6.3) User Add Post or Entry


User logins to CampusConnect
User opens addition page
[If user prefers add entry]
[If user prefers add post]
User adds entry
User adds post
User fills blanks (title,description,price,category,subcategory)
[If user do not prefer add an image]
[If user prefer add an]
User publishes post or entry

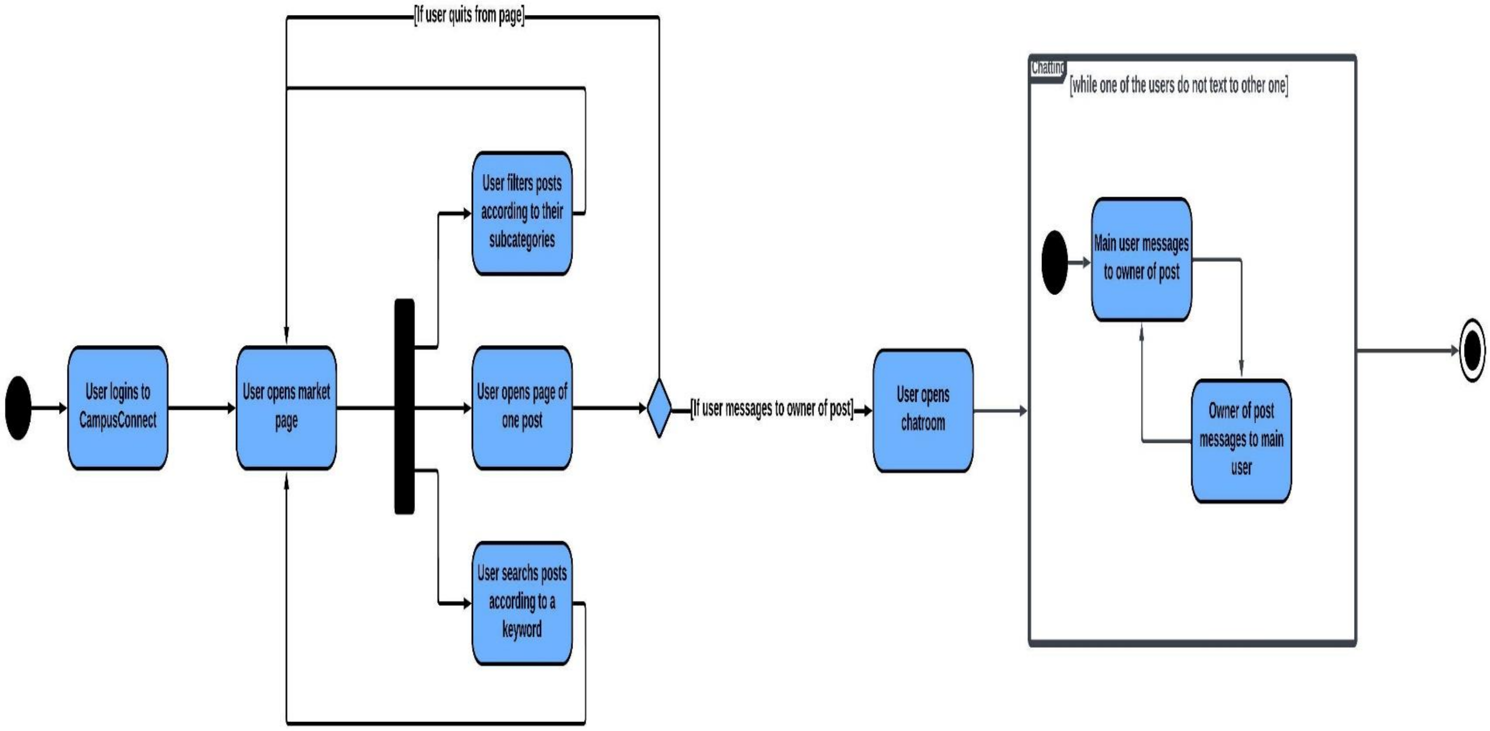
6.4) User Uses Borrowing Feature


User logins to CampusConnect
User opens borrowing page
User filters borrowings according to their subcategories
User opens add borrowing page
User publishes borrowing ad
User opens page of one borrowing
[If user quits from the page]
[If user messages to owner of borrowing]
User opens chatroom
User searches borrowings according to a keyword
Chatting
[while one of the users do not text to other one]
Main user messages to owner of post
Owner of post messages to main user

6.5) User Uses Forum Feature

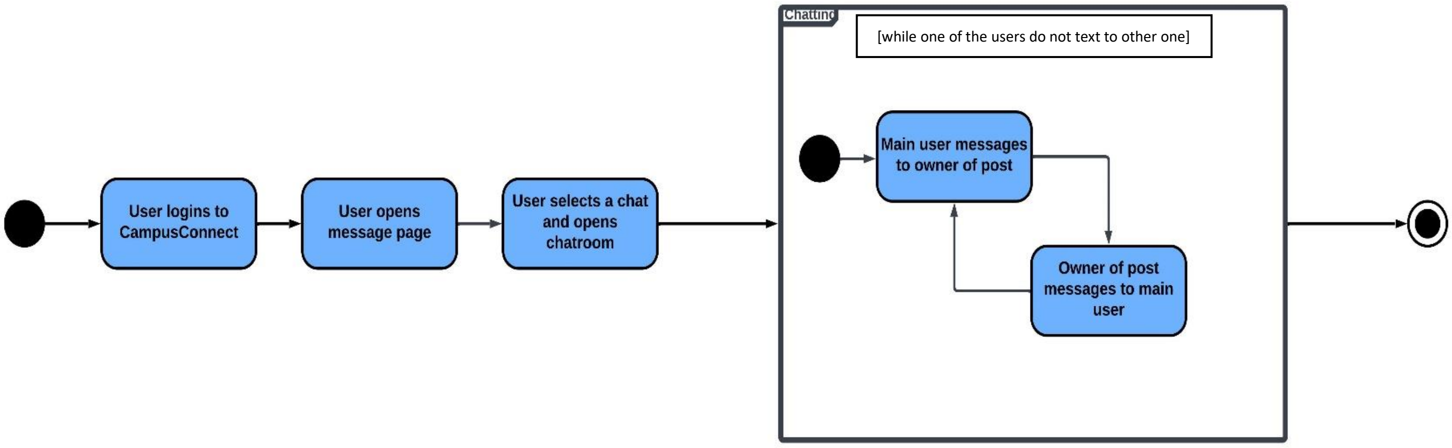

[If user quits from the page]
User logins to CampusConnect
User opens forum page
User filters entries according to their subcategories
User opens page of one entry
User searches entries according to keyword
[If user comments to the entry]

6.6) User Uses Market Feature


[If user quits from page]
User logins to CampusConnect
User opens market page
User filters posts according to their subcategories
User opens page of one post
User searchs posts according to a keyword
[If user messages to owner of post]
User opens chatroom
Chatting
[while one of the users do not text to other one]
Main user messages to owner of post
Owner of post messages to main user

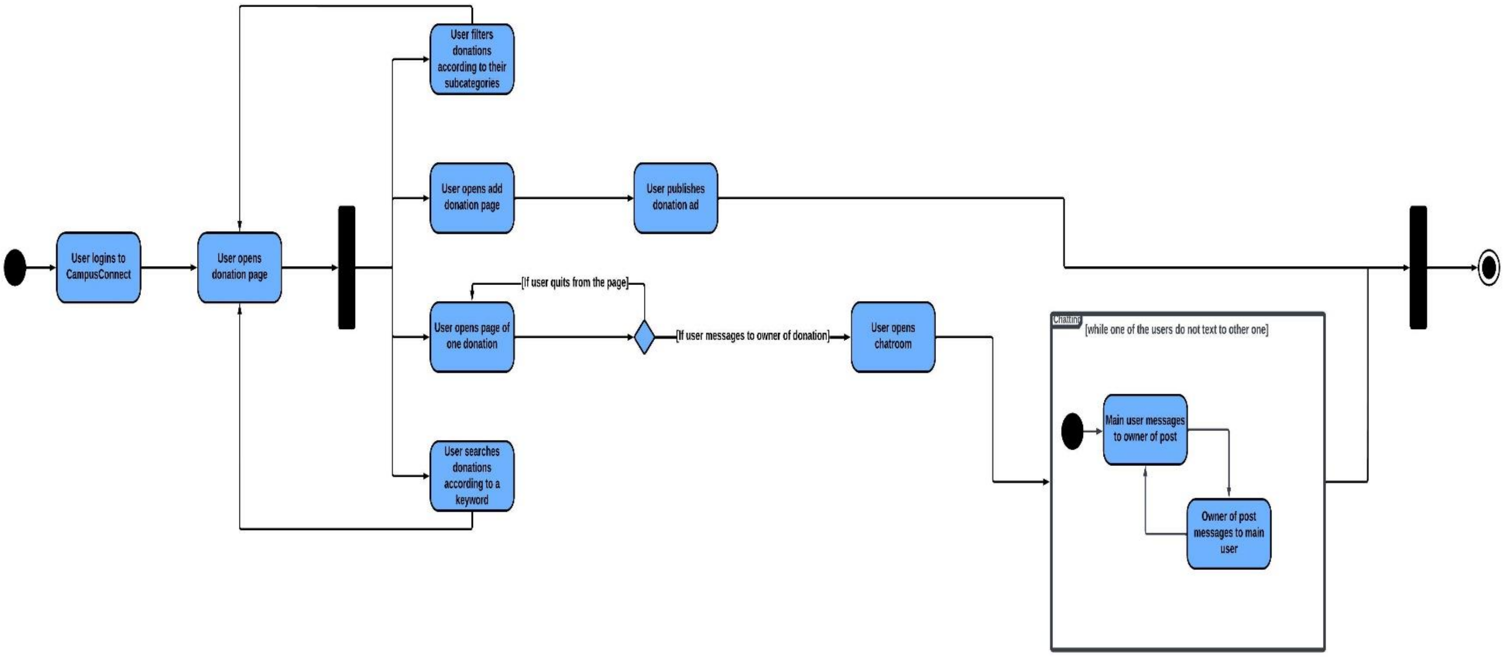
6.7) User Uses Message Feature

[while one of the users do not text to other one]



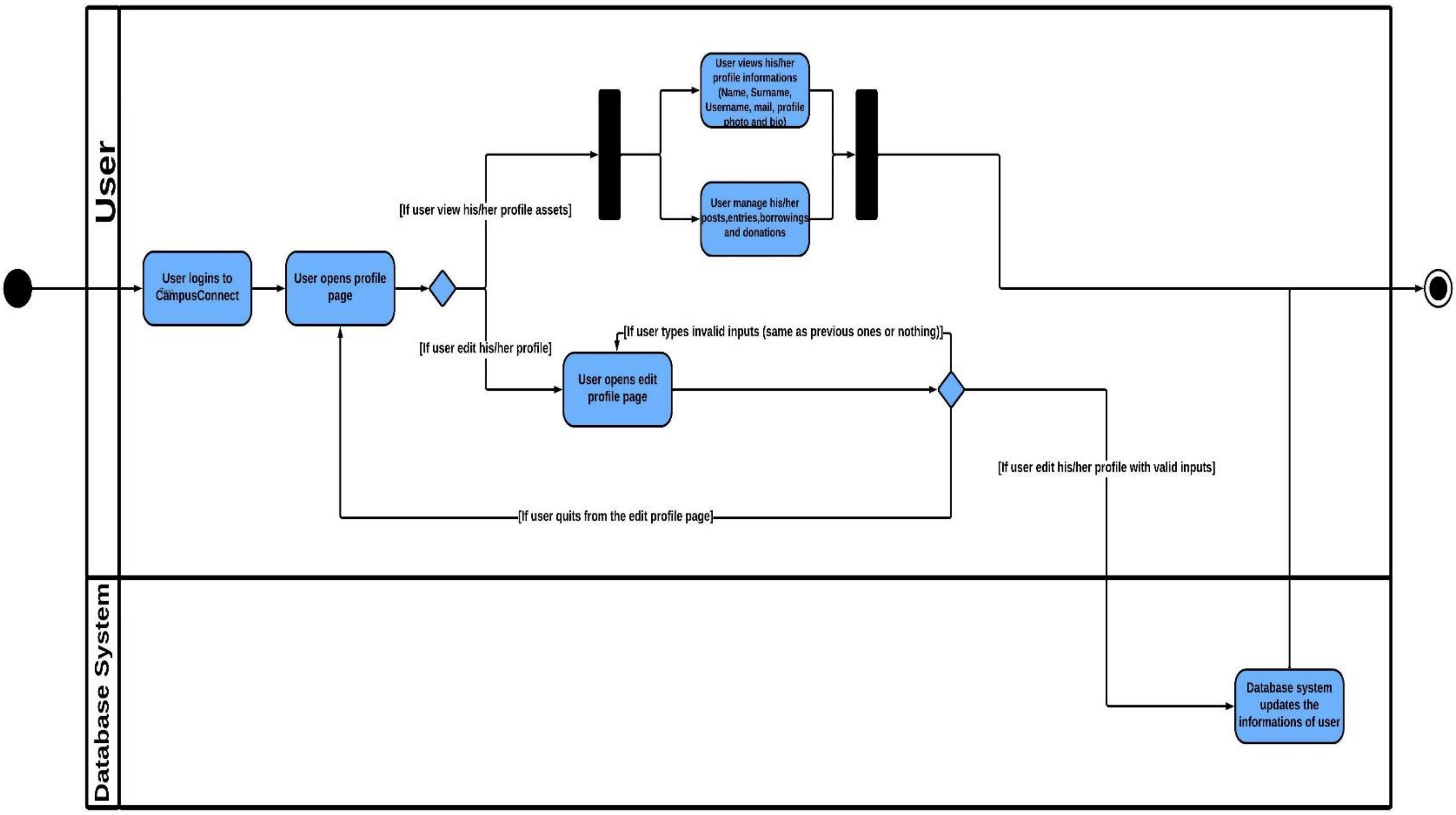
User logins to CampusConnect
User opens message page
User selects a chat and opens chatroom
Chatting
[while one of the users do not text to other one]
Main user messages to owner of post
Owner of post messages to main user

6.8) Users Uses Donation Feature



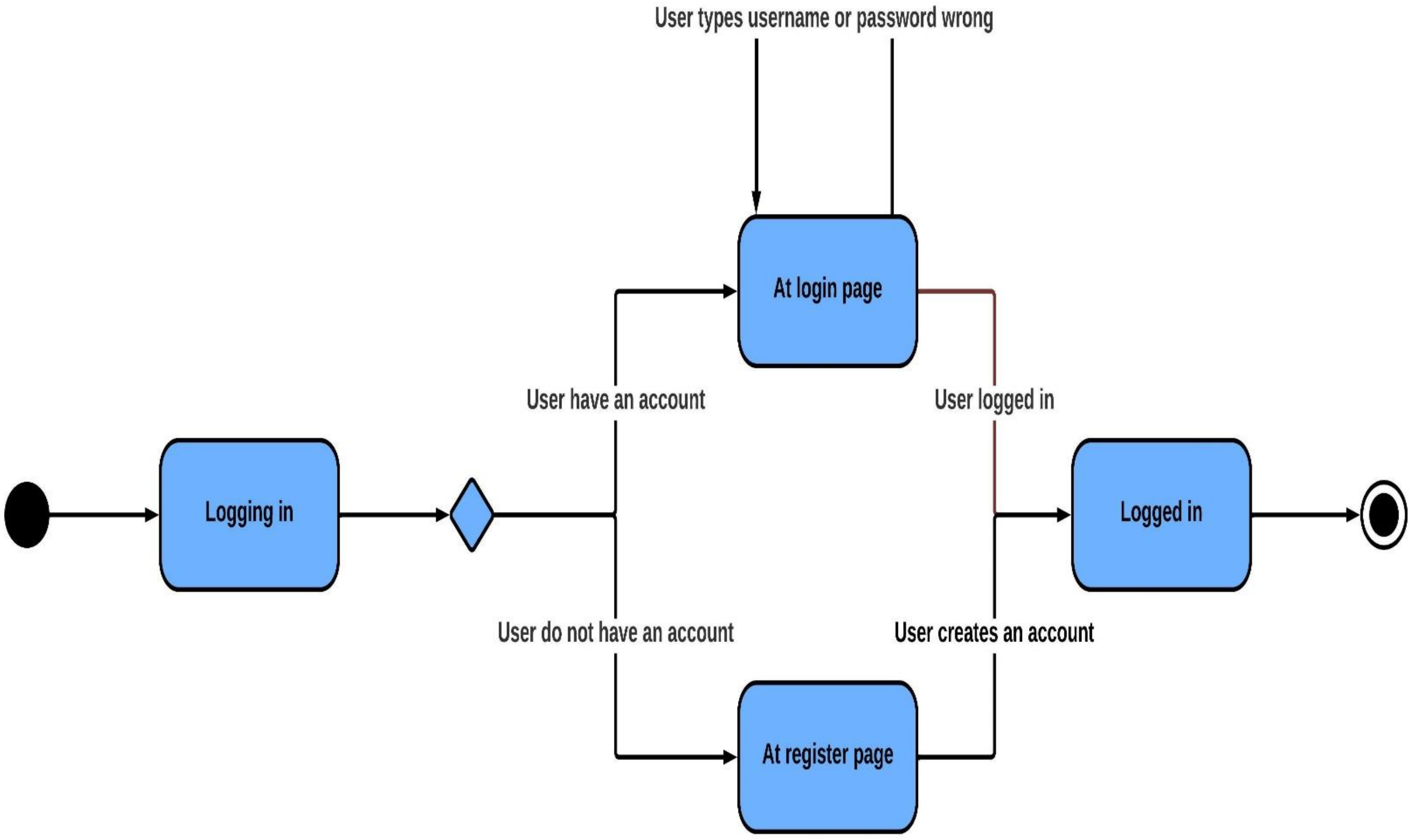
User logins to CampusConnect
User opens donation page
User filters donations according to their subcategories
User opens add donation page
User publishes donation ad
User opens page of one donation
[If user quits from the page]
[If user messages to owner of donation]
User opens chatroom
User searches donations according to a keyword
Chatting
[while one of the users do not text to other one]
Main user messages to owner of post
Owner of post messages to main user

6.9) Users Uses Features of Profile Page

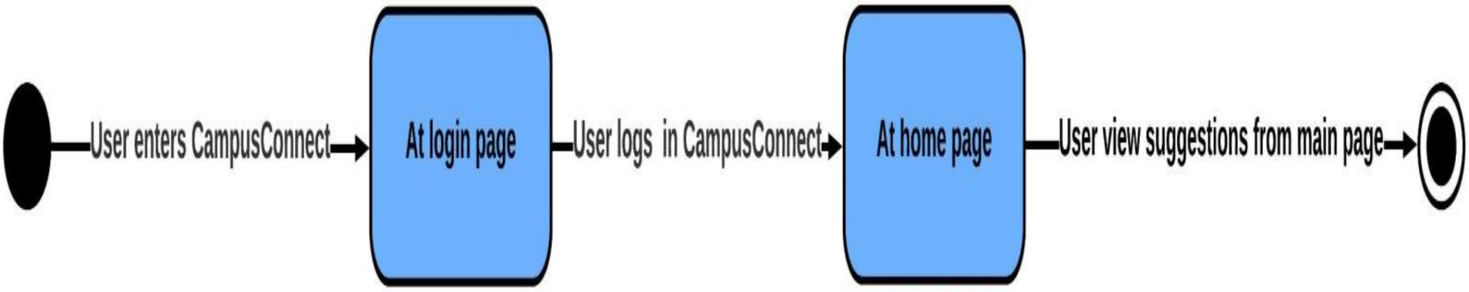

User
Database System
User logins to CampusConnect
User opens profile page
[If user view his/her profile assets]
User views his/her profile informations (Name, Surname, Username, mail, profile photo and bio)
User manage his/her posts,entries,borrowings and donations
[If user edit his/her profile]
[If user types invalid inputs (same as previous ones or nothing)]
User opens edit profile page
[If user edit his/her profile with valid inputs]
[If user quits from the edit profile page]
Database system updates the informations of user

7) State Diagrams

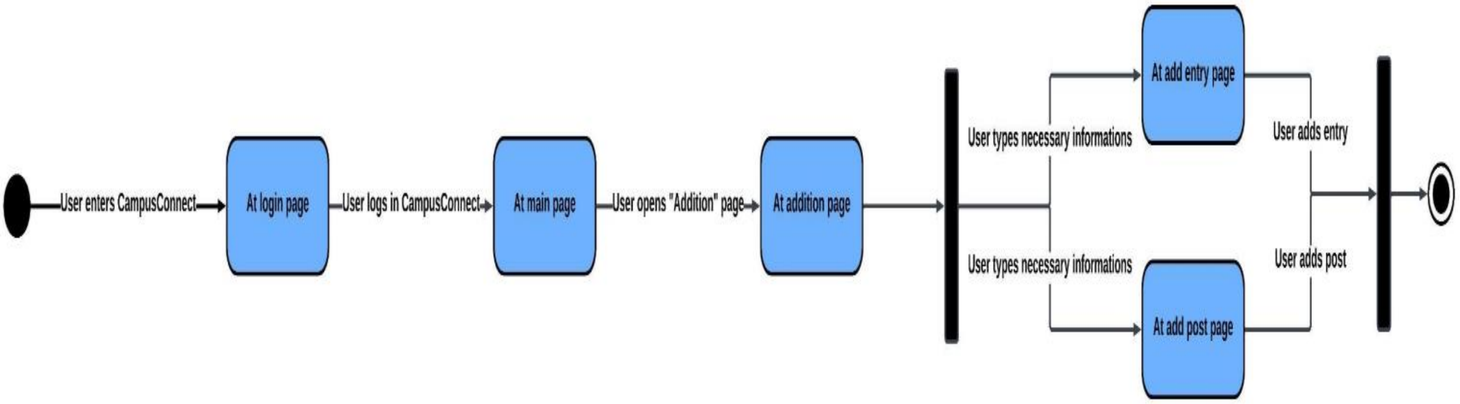
7.1) User Logins Into his/her Account



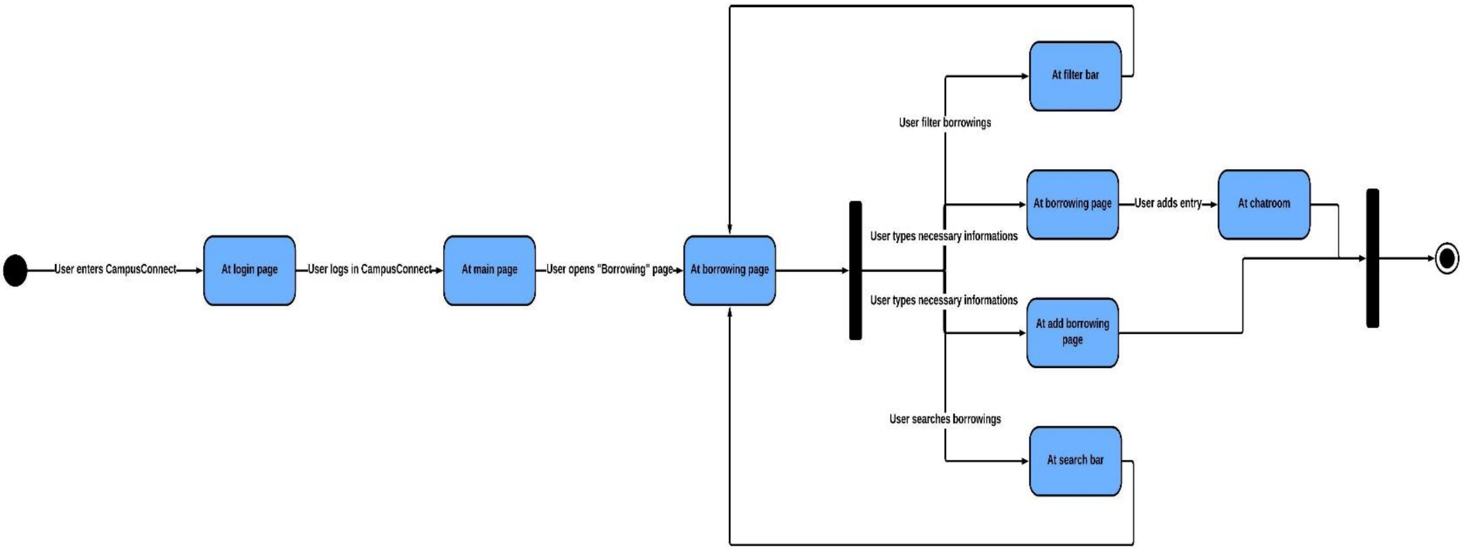
7.2) User Views Suggestions at Main Page


User enters CampusConnect
At login page
User logs in CampusConnect
At home page
User view suggestions from main page

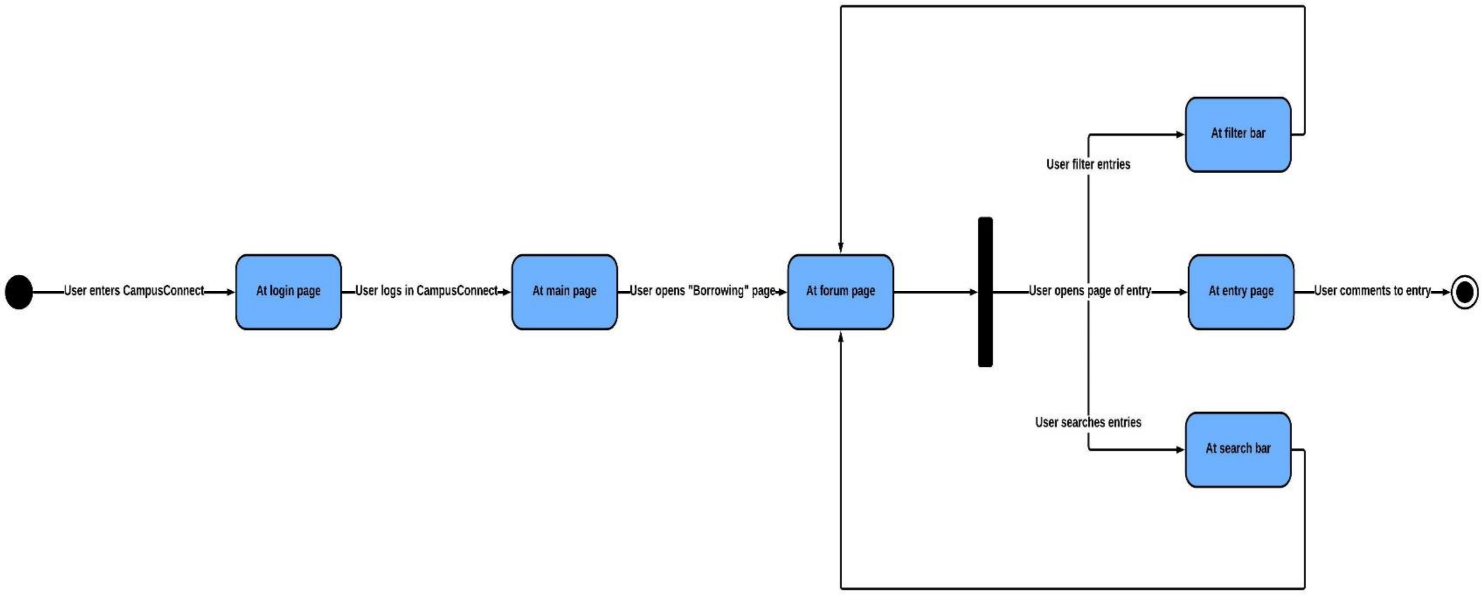
7.3) User Add Post or Entry


User enters CampusConnect
At login page
User logs in CampusConnect
At main page
User opens "Addition" page
At addition page
User types necessary informations
At add entry page
User adds entry
User types necessary informations
At add post page
User adds post

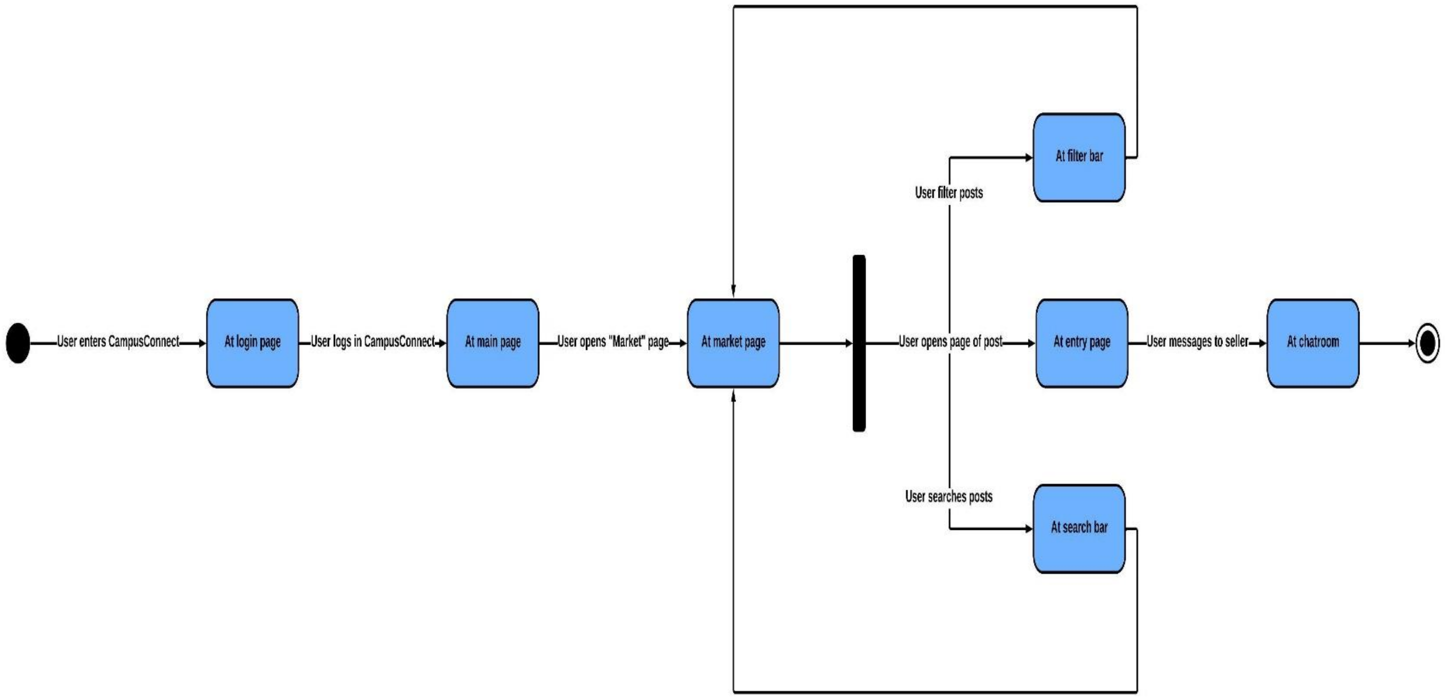
7.4) User Uses Borrowing Feature



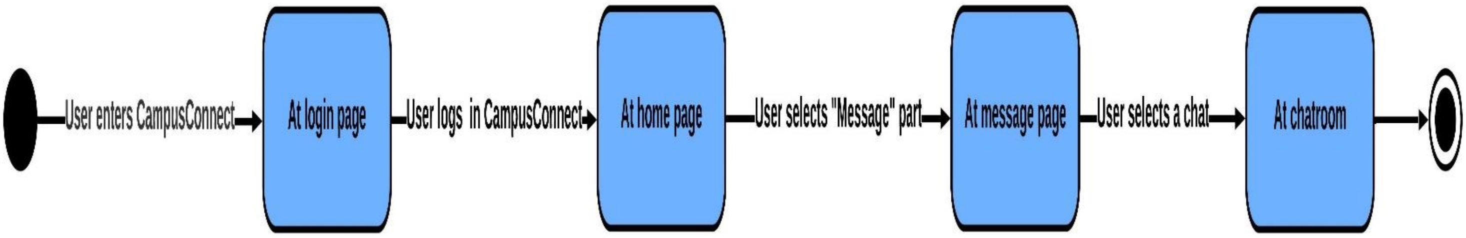
7.5) User Uses Forum Feature



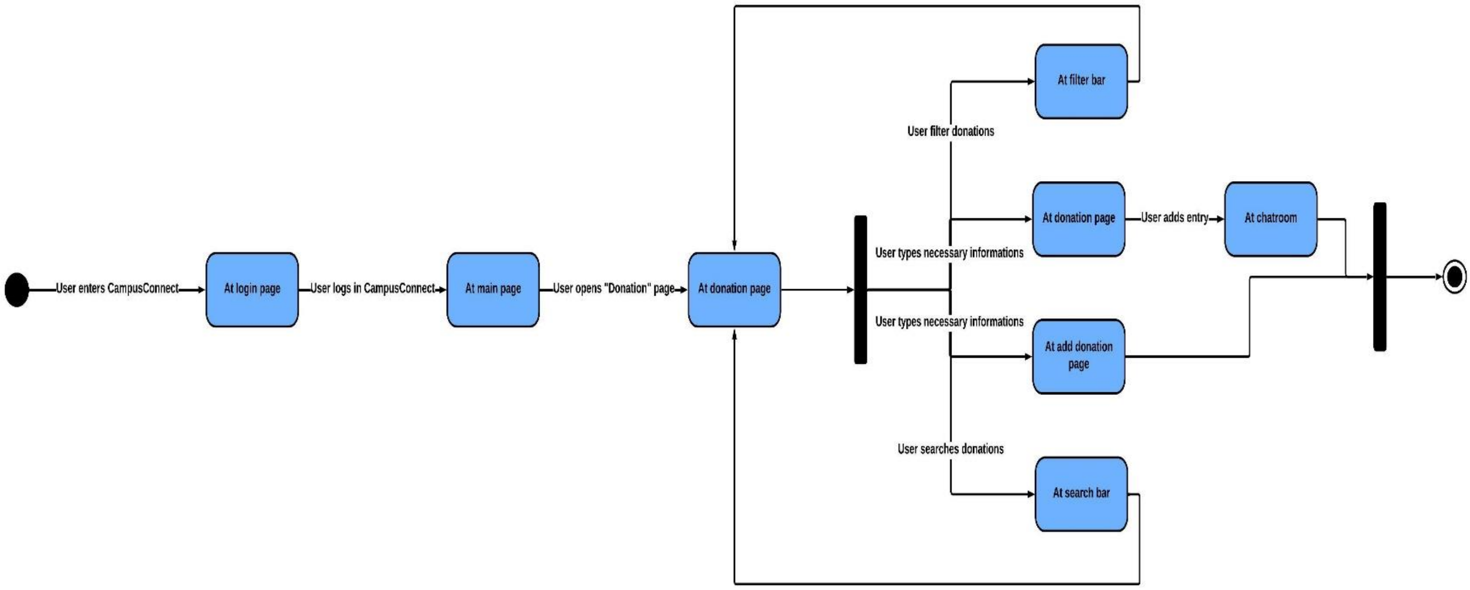
7.6) User Uses Market Feature



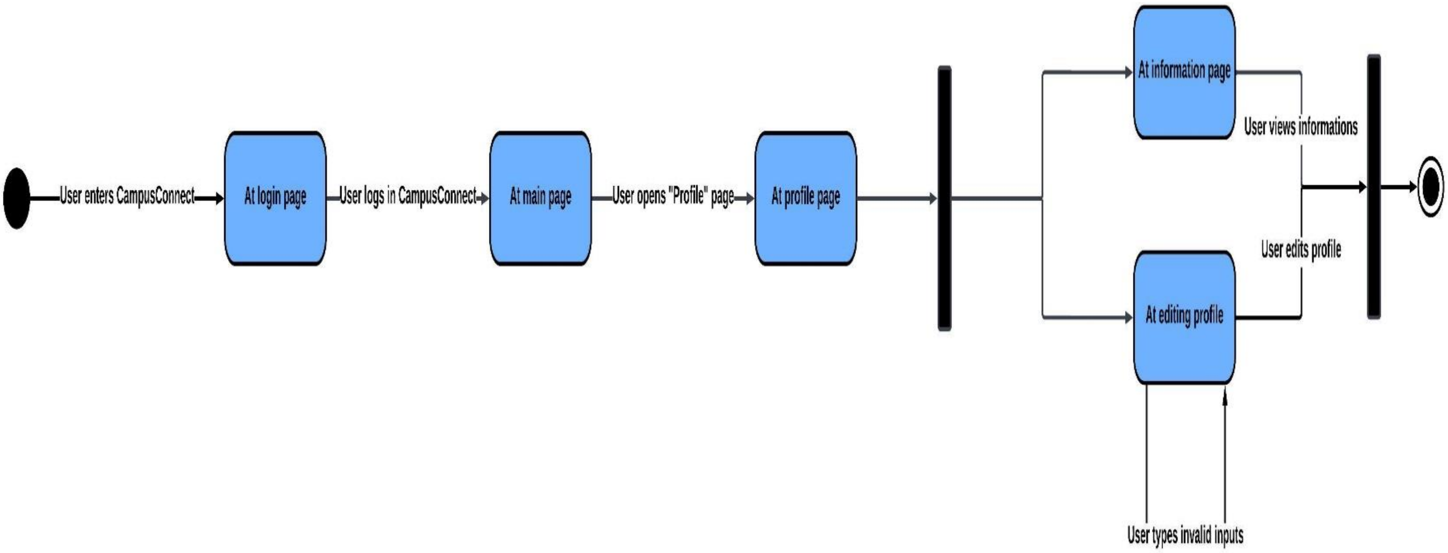
7.7) User Uses Message Feature


User enters CampusConnect
At login page
User logs in CampusConnect
At home page
User selects "Message" part
At message page
User selects a chat
At chatroom

7.8) Users Uses Donation Feature

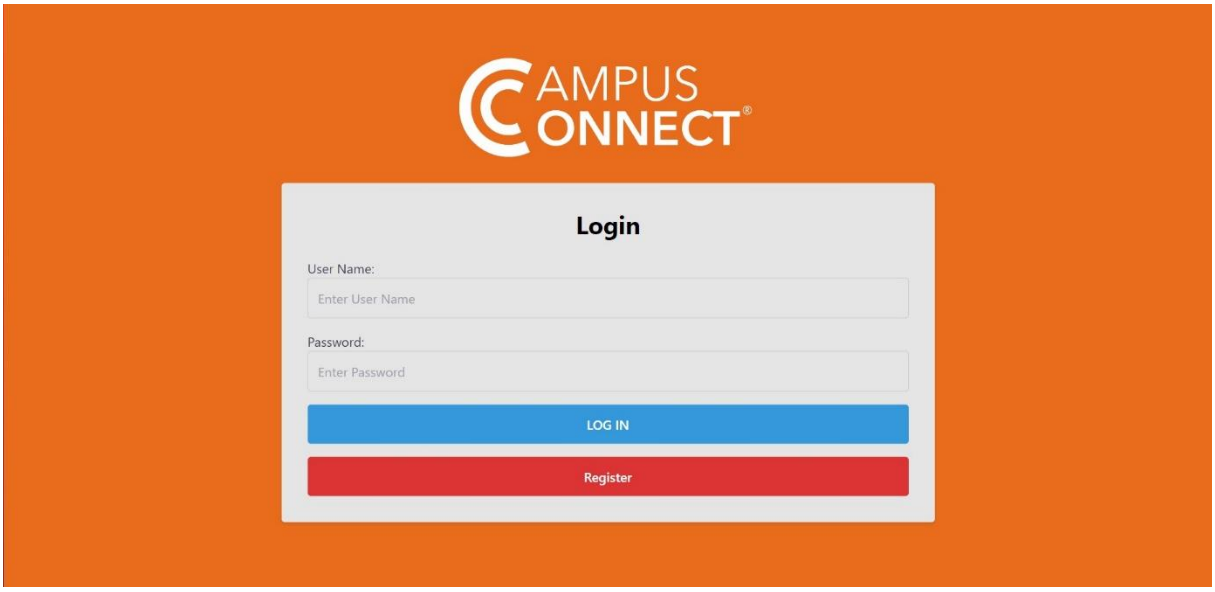


7.9) Users Uses Features of Profile Page

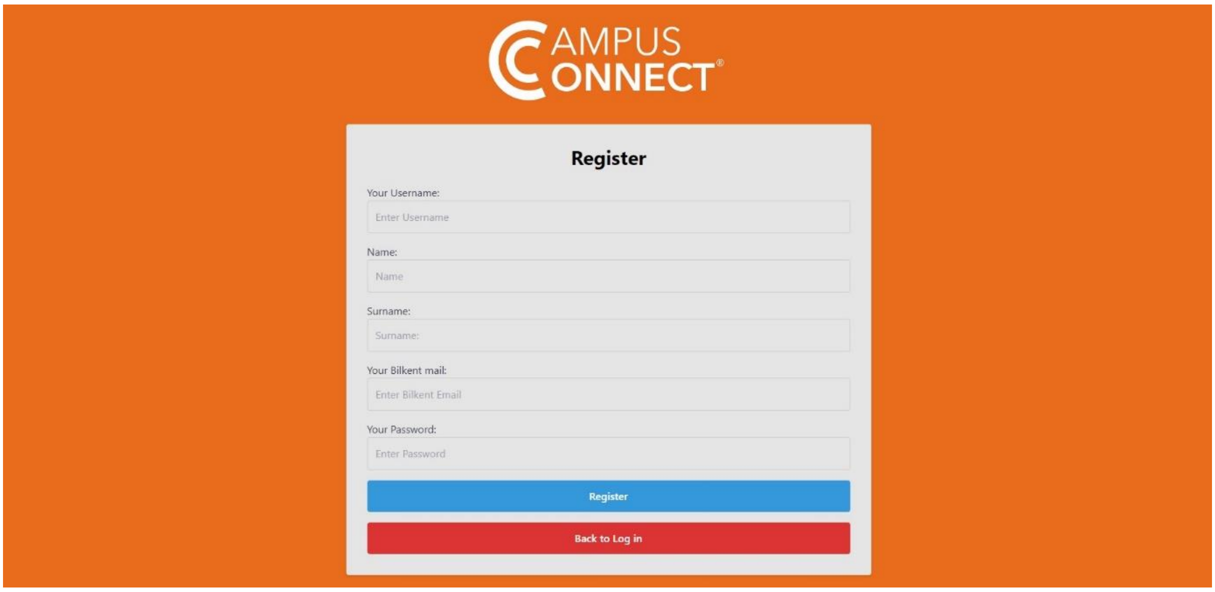


8) Mockups

8.1) Login Page



8.2) Register Page



Register

Your Username:

Enter Username

Name:

Name

Surname:

Surname:

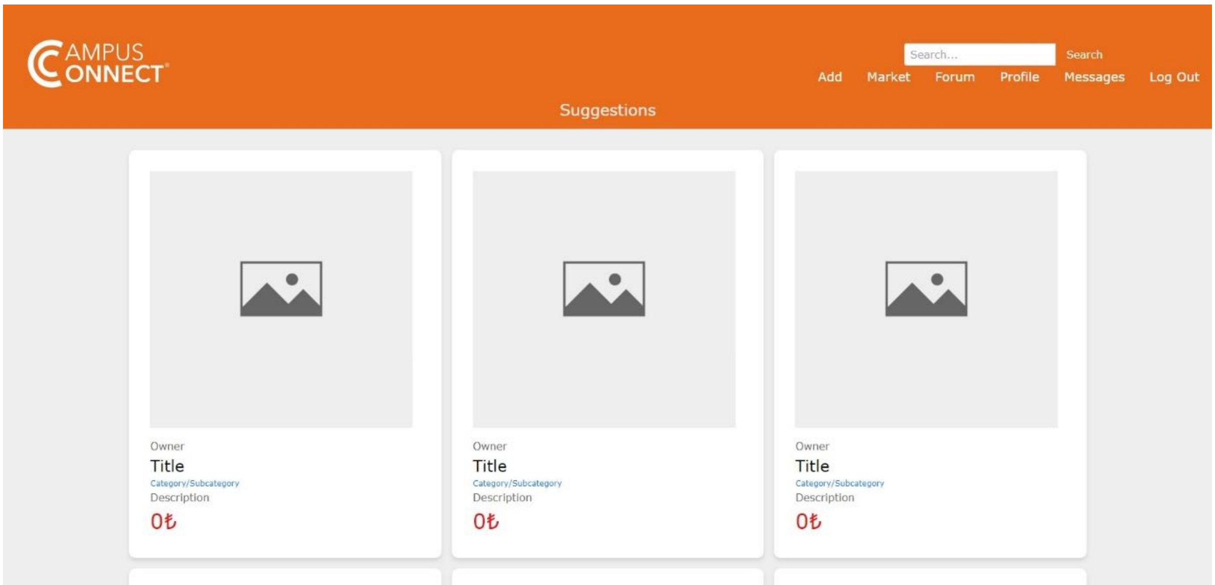
Your Bilkent mail:

Enter Bilkent Email

Your Password:

Enter Password

8.3) Home Page

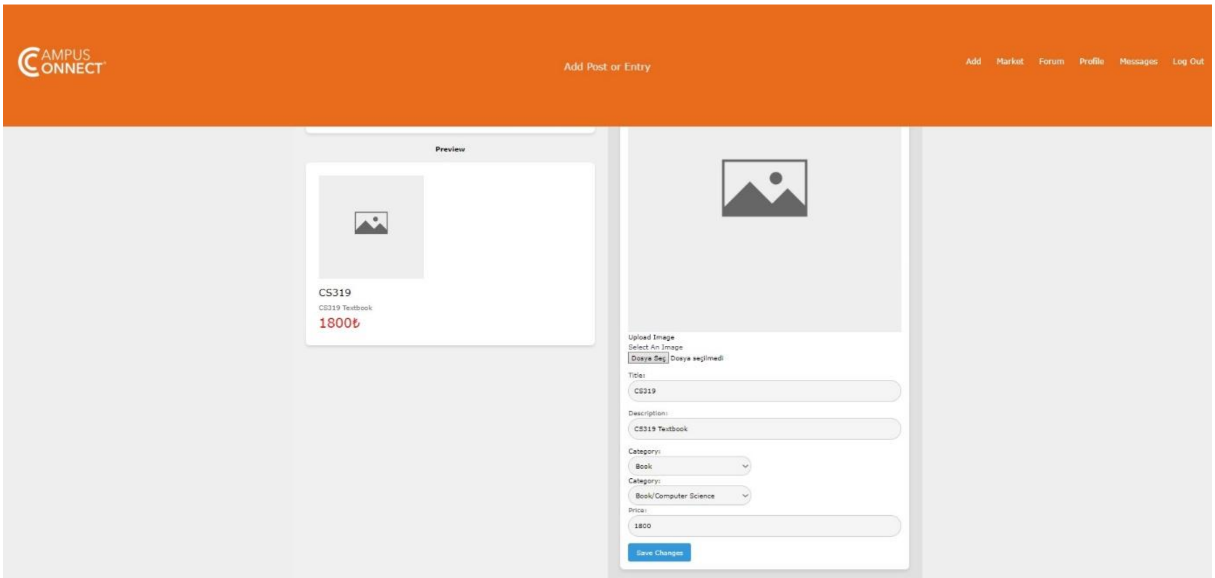


Owner
Title
Category/Subcategory
Description
0₺

Owner
Title
Category/Subcategory
Description
0₺

Owner
Title
Category/Subcategory
Description
0₺

8.4) Add Page



CS319
CS319 Textbook
1800₺

Dosya seçilmedi

Title: CS319

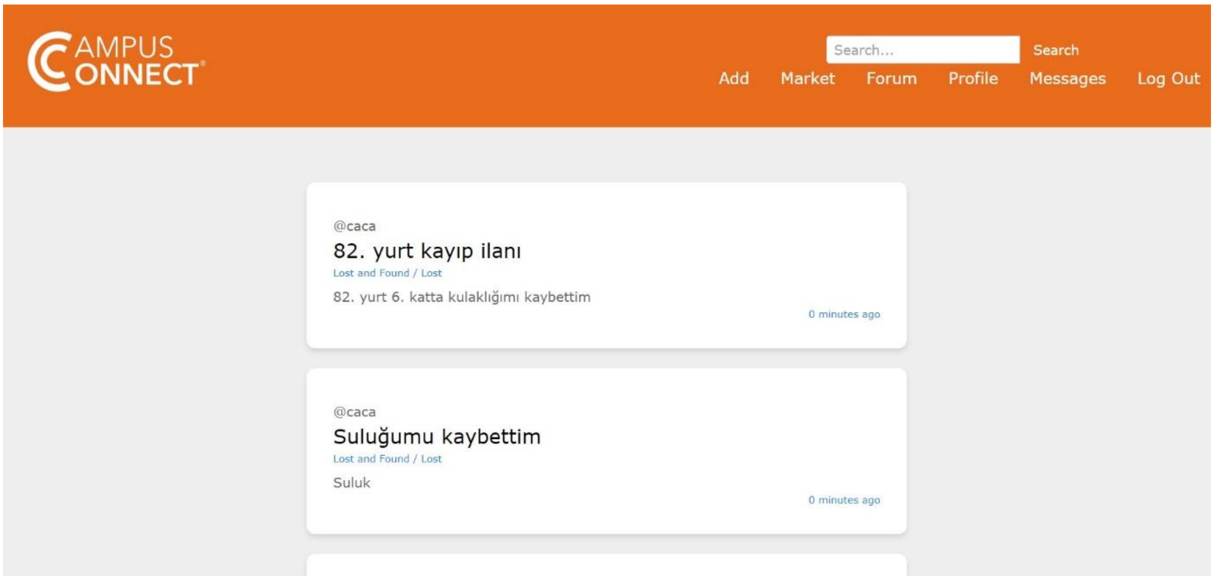
Description: CS319 Textbook

Category: Book

Category: Book/Computer Science

Price: 1800

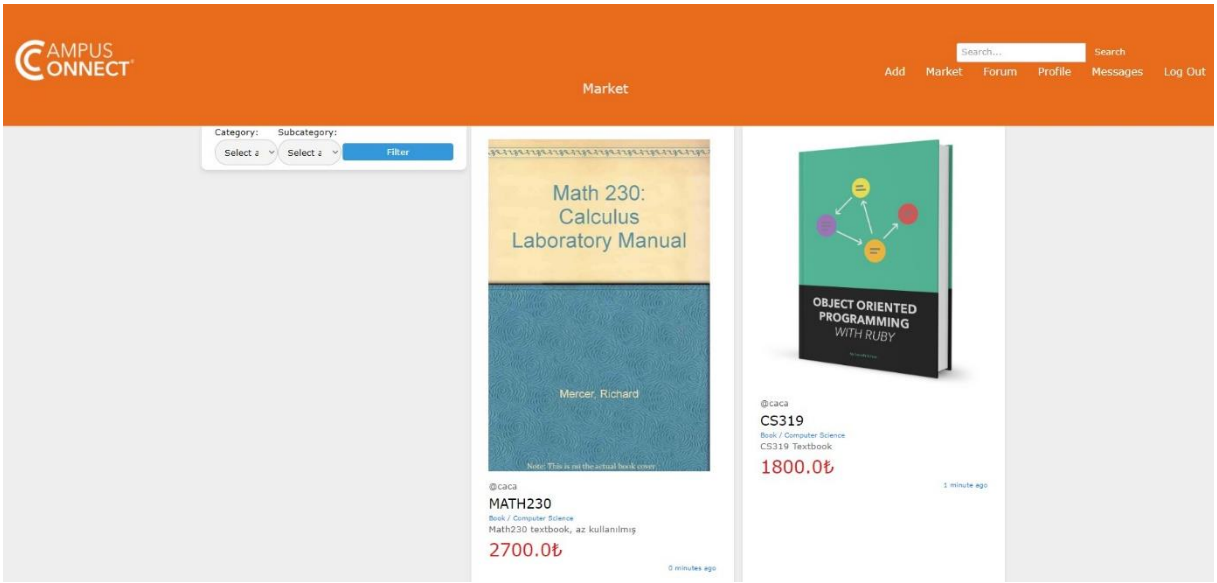
8.5) Market Page



@caca
82. yurt kayıp ilanı
Lost and Found / Lost
82. yurt 6. katta kulaklığımı kaybettim
0 minutes ago

@caca
Suluğumu kaybettim
Lost and Found / Lost
Suluk
0 minutes ago

8.6) Forum Page



Math 230: Calculus Laboratory Manual

Mercer, Richard

@caca

MATH230

Book / Computer Science

Math230 textbook, az kullanılmış

2700.0₺

0 minutes ago

OBJECT ORIENTED PROGRAMMING WITH RUBY

@caca

CS319

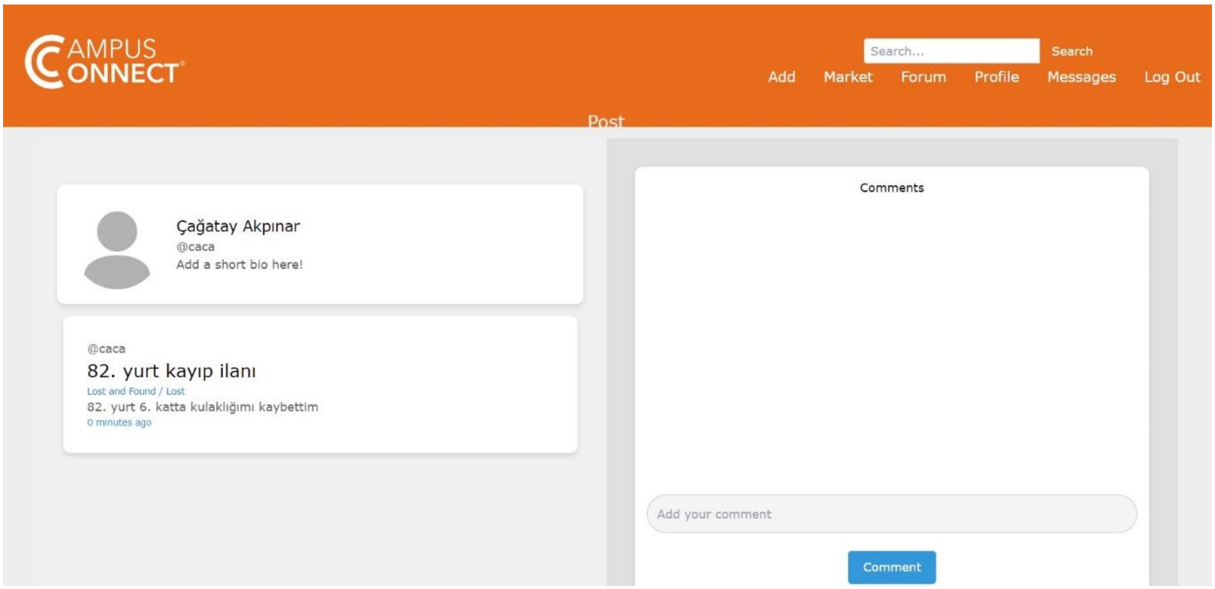
Book / Computer Science

CS319 Textbook

1800.0₺

1 minute ago

8.7) Entry Page



Çağatay Akpınar

@caca

Add a short bio here!

@caca

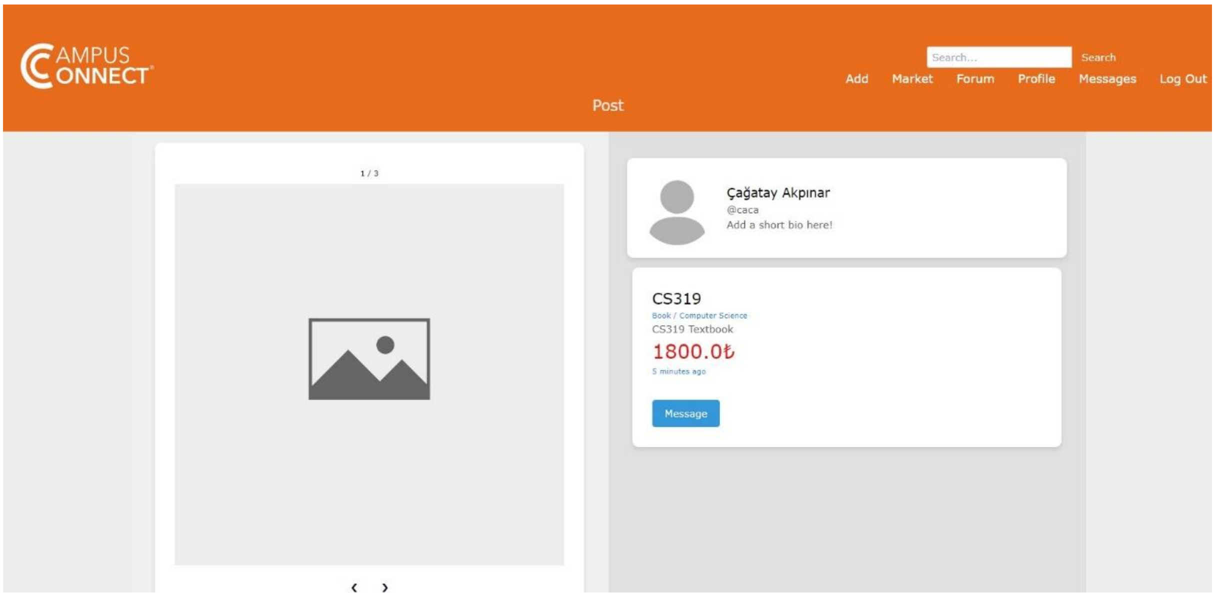
82. yurt kayıp ilanı

Lost and Found / Lost

82. yurt 6. katta kulaklığımı kaybettim

0 minutes ago

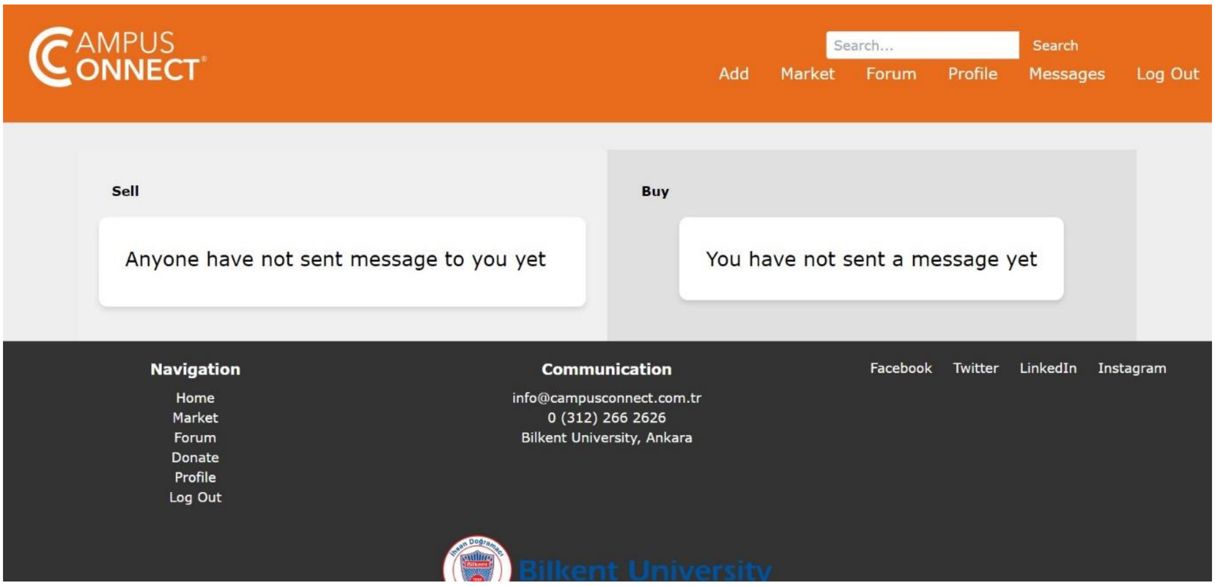
8.8) Post Page



Çağatay Akpınar
@caca
Add a short bio here!

CS319
Book / Computer Science
CS319 Textbook
1800.0₺
5 minutes ago

8.9) Message Page



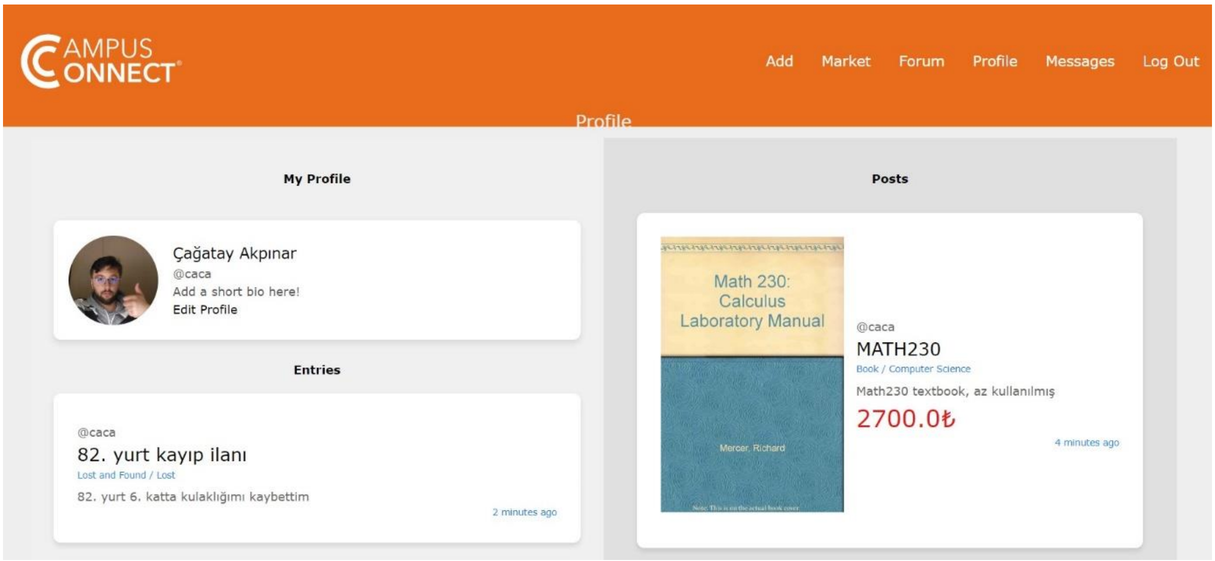
Sell

Anyone have not sent message to you yet

Buy

You have not sent a message yet

8.10) Profile Page



My Profile

Çağatay Akpınar
@caca
Add a short bio here!
Edit Profile

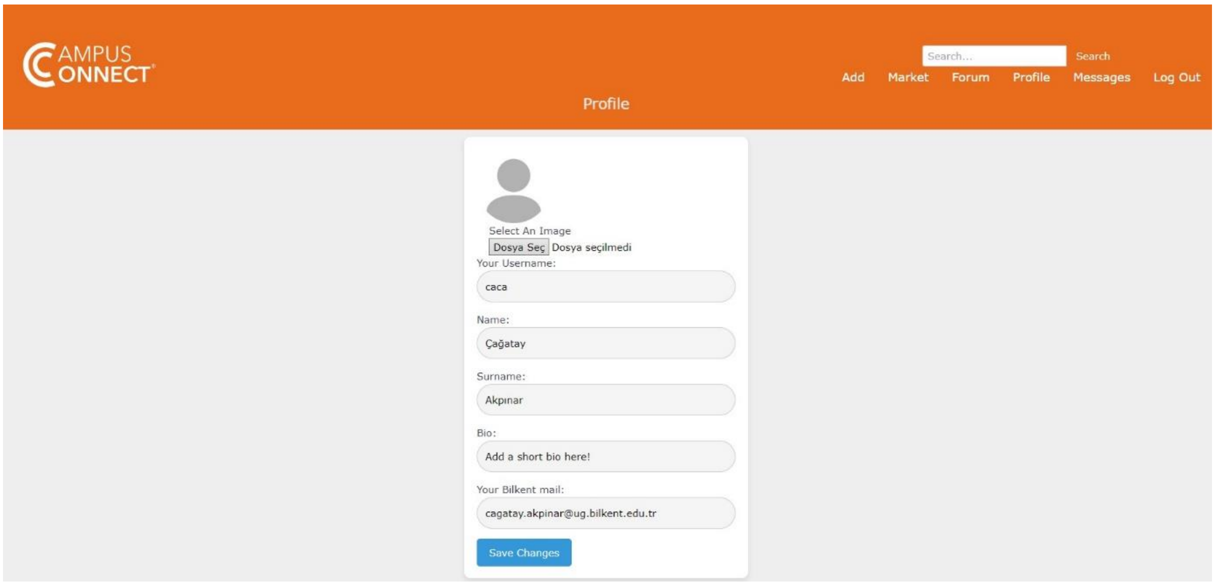
Entries

@caca
82. yurt kayıp ilanı
Lost and Found / Lost
82. yurt 6. katta kulaklığımı kaybettim
2 minutes ago

Posts

@caca
MATH230
Book / Computer Science
Math230 textbook, az kullanılmış
2700.0₺
4 minutes ago

8.11) Edit Profile Page



Dosya seçilmedi

Your Username: caca

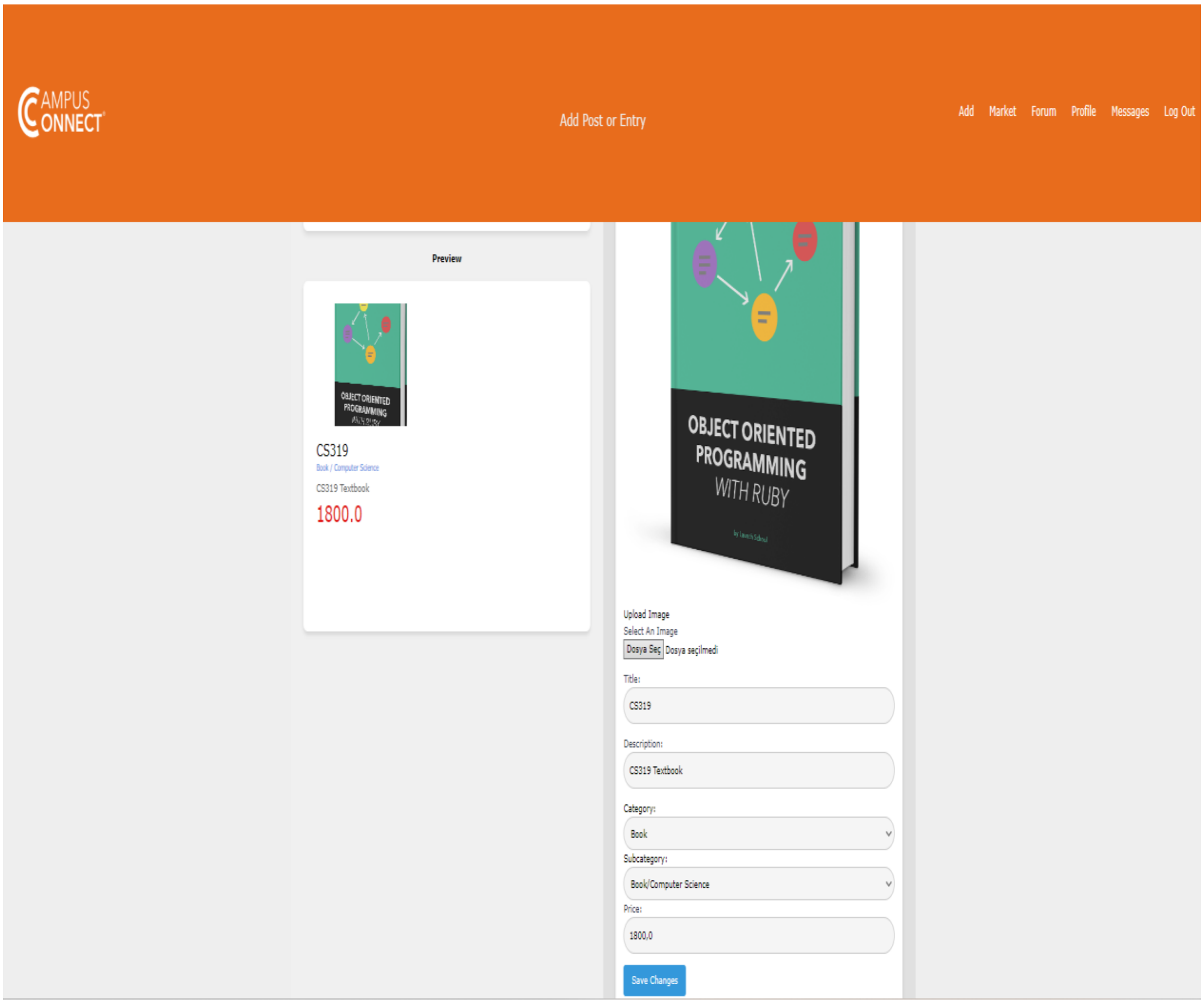
Name: Çağatay

Surname: Akpınar

Bio: Add a short bio here!

Your Bilkent mail: cagatay.akpinar@ug.bilkent.edu.tr

8.12) Edit Post Page



CS319

Book / Computer Science

CS319 Textbook

1800.0

Upload Image

Select An Image

Dosya Seç Dosya seçilmedi

Title: CS319

Description: CS319 Textbook

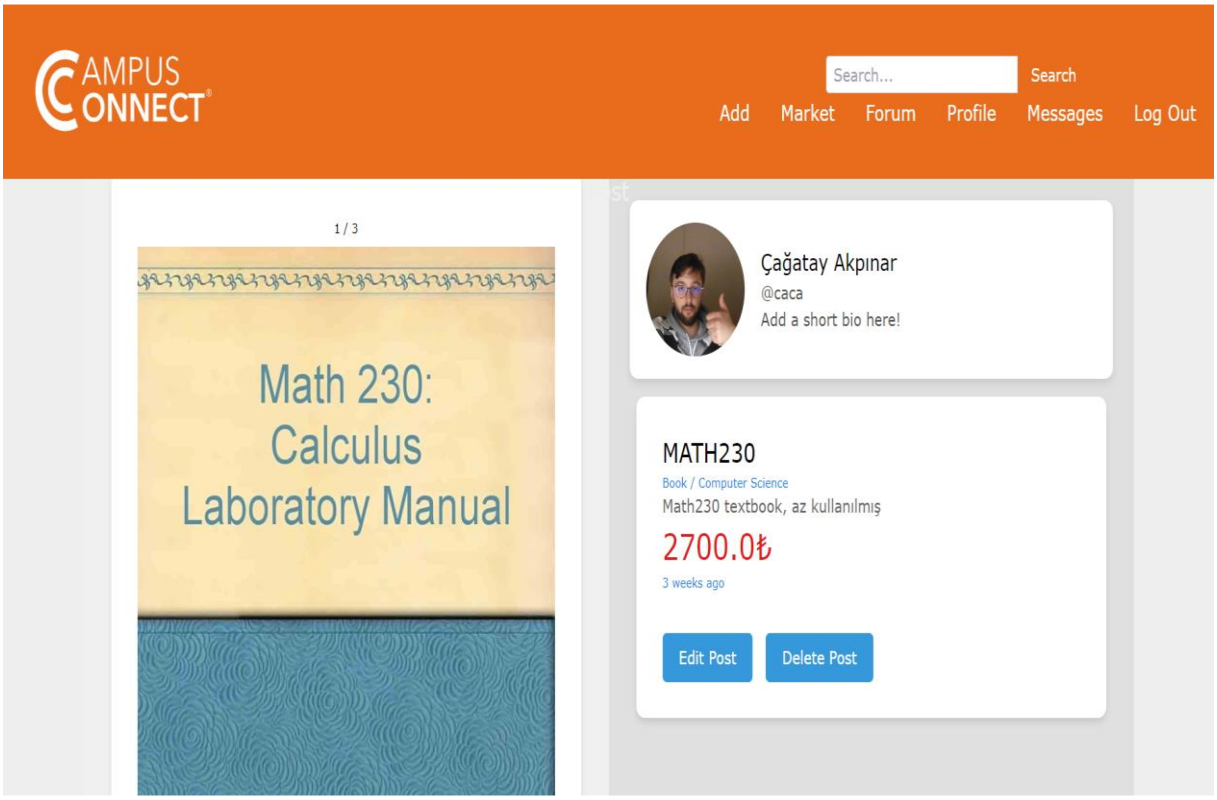
Category: Book

Subcategory: Book/Computer Science

Price: 1800.0

Save Changes

8.13) Delete and Edit Post Buttons Page



Çağatay Akpınar
@caca
Add a short bio here!

MATH230
Book / Computer Science
Math230 textbook, az kullanılmış
2700.0₺
3 weeks ago