Gear4Camp - Outdoor Gear Rental - Iteration 2



September 25th, 2016

**Table of Contents**

1. Introduction
2. Supplementary Specification
3. Glossary
4. Artifacts
5. Requirements
6. Domain Model
7. Consolidated Usecase Diagram
8. Consolidated Sequence Diagram
9. Class Diagram
10. Use Case Models
11. Operation Contracts
12. Database Schema
13. Testing
14. Project Management

**1. Introduction**

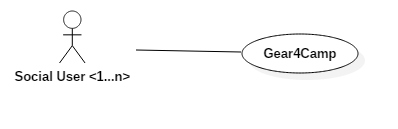
Gear4Camp is an a website which acts as online rental system for nearby users to rent outdoor equipment to each other conveniently.

In our website we will be showing following features

* Users can register for an account and become a member.
* Members can create and post Ads.
* Members can edit or delete existing Ads.
* Members can rent an item.
* Members can rate their rental transactions

**2. Supplementary Specification**

**System Perspective**



**Assumptions and Dependencies**

* Users must be able to view all available ads.
* Clicking an ad will bring the user to a new page to display more details about the ad.
* Online Payment and shipping an item are outside the scope of our system.
* Renters and buyers will contact each other and decide on payment and pick up or drop location.

**Stakeholder Perspective**

We have only one stakeholder which is common user who will be posting rent item Ads or renting items from other users.

**Tools Used**

* GitHub: Source code and documentation repository
* Eclipse: Development environment
* Trello: Project tracking and management
* Google Docs: Document Collaboration
* UML Design: StarUML

**Technologies Used**

* Java, JSP, Servlets, HTML, CSS
* Server: Apache Tomcat
* Database: MySql

**3. Glossary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Version** | **Date** | **Description** | **Author** |
| V 1.0 | 09/25/2016 | First draft to be refined in later versions | Group 3 |
| V 2.0 |  |  | Group 3 |

|  |  |
| --- | --- |
| **Term** | **Description** |
| Guest | Someone who uses the Gear4Camp website without creating an account. I.e. they can view and choose ads that they may be interested in and contact the owner, but cannot create ads themselves or rent an item through the website. |
| Member | A user who creates an account and can create ad postings to rent out equipment. |
| User | A person that uses the website in general. A user is not particularly a guest or a member. |
| Ad | Created by a member to advertise equipment available to rent. It includes various details about the equipment and various policies for renting (dates, price, etc.). |
| Home Page | Home page of the website that displays all of the ads created by users. It also contains a search bar and a side bar for various categories. |

**4. Artifacts**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Discipline** | **Artifact** | **Inception (Iteration 1)** | **Elaboration (Iteration 1)** | **Construction (Iteration2 )** | **Transition (Iteration)** |
| Business Modelling | Domain Model |  | S, R | R |  |
| Requirements | Use Case Model |  | S | R |  |
|  | Use Case Text | S | R | R |  |
|  | Use Case Diagram | S | R | R |  |
|  | System Sequence Diagram |  | S | R |  |
|  | Vision | S | R |  |  |
|  | Supplementary Specification |  | S |  |  |
|  | Glossary |  | S |  |  |
| Design | Class Diagram |  | S | R |  |
|  | Interaction Diagram |  | S |  |  |
|  | Sequence Diagram |  | S | R |  |
| Implementation | Code |  | S | R |  |
| Testing | Test Case Plan |  | S |  |  |
|  | Unit Testing |  | S | R |  |
|  | System Testing |  | S | R |  |

**5. Requirements**

**R1 -** Guests shall be able to sign up to become members. The member account will have typical information such as name, email address, general location, contact number, etc. An automatically generated User ID number will be given to the account upon creation.

**R2** - Members shall be able to log into their accounts.

**R3** - Members shall be able to create an ad with equipment information, 1 required picture maximum only for now, required email address for contact, optional phone number for additional contact, date ranges for renting, the renting location, and requested price per day to rent. Ads will have an automatically generated and unique Ad ID number.

**R4** - All users shall be able to browse equipment for rental ads on the home page.

**R5** - Members shall be able to view and manage their accounts on the account page and can view their ad posting history.

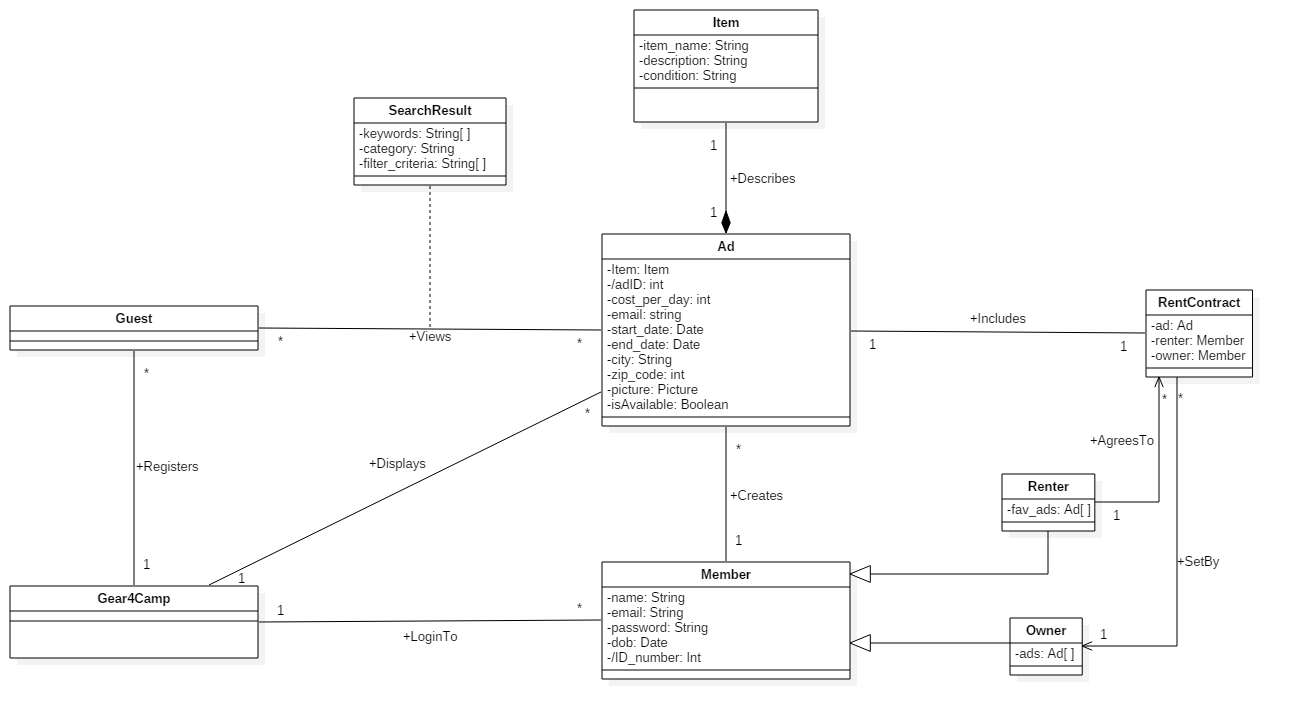
**R6** - Members shall be able to manage their ad postings.

**R7** - Users shall be able to find ads for specific equipment by searching based on keywords, using categories, or filtering based on specific criteria.

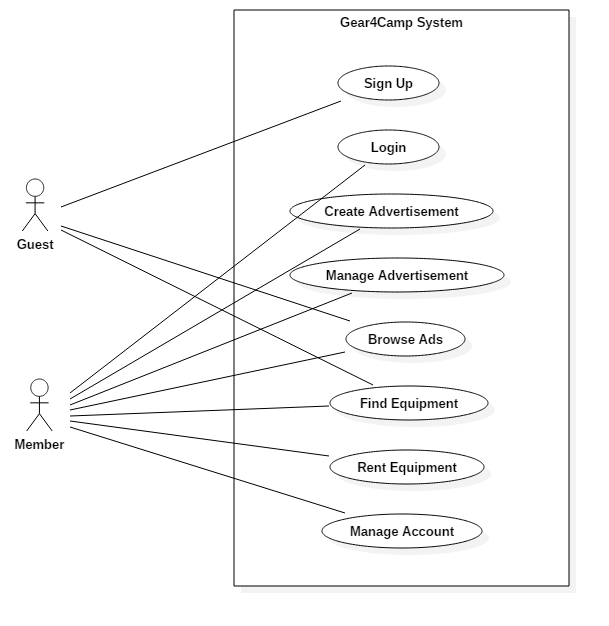
**R8** - Members shall be able to rent equipment through the website.

|  |  |
| --- | --- |
| **Requirement and Use Case Mapping** | |
| **Requirement** | **Use Case** |
| **R1** | Use Case UC1: Sign Up |
| **R2** | Use Case UC2: Login |
| **R3** | Use Case UC3: Create Ad |
| **R4** | Use Case UC4: Browse Ads |
| **R5** | Use Case UC5: Manage Account |
| **R6** | Use Case UC6: Manage Ad |
| **R7** | Use Case UC7: Find Ads |
| **R8** | Use Case UC8: Rent Equipment |

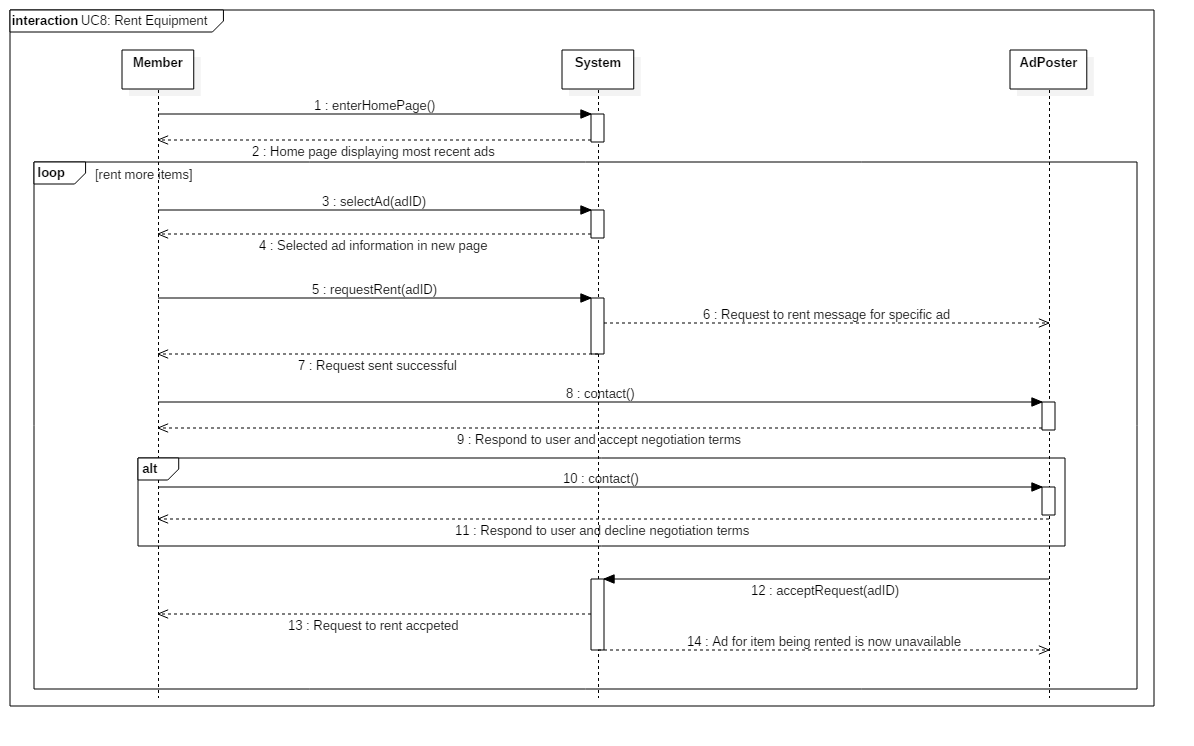
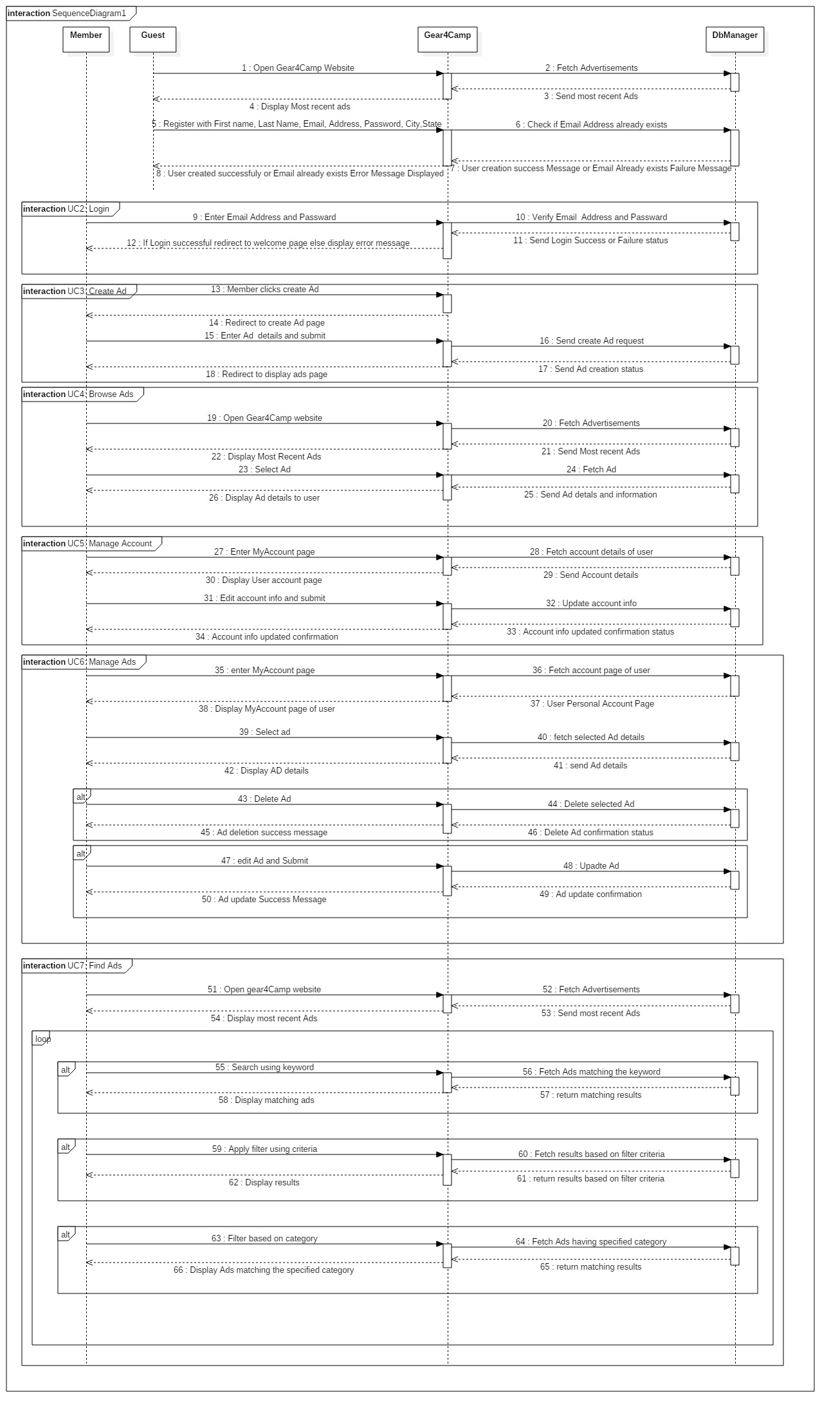
**6. Domain Model**



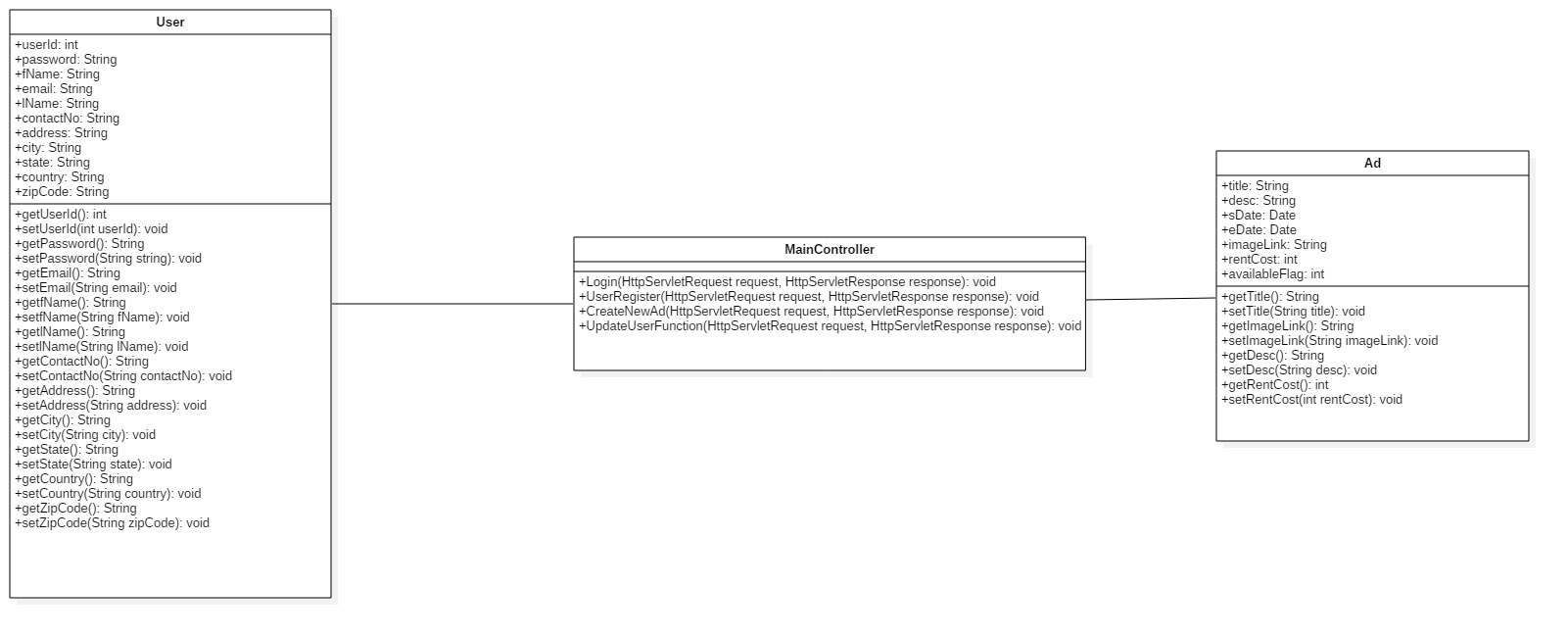
**7. Consolidated Use Case Diagram**



**8. Consolidated Sequence Diagram**



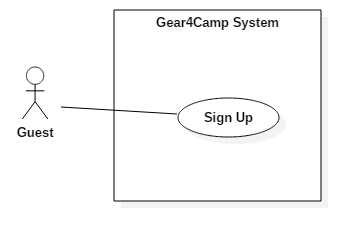
**9. Class Diagram**



**10. Use Case Model**

**Use Case UC1: Sign Up**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (primary)

**Stakeholders and Interests:**

* User**:** wants to register for a member account.

**Main Success Scenario:**

1. User opens the website.
2. User selects the link to register for an account.
3. System prompts user for registration information such as name, email address used as username, password, phone number, state, address, city and zip code.
4. User enters his/her registration information and submits the details.
5. System verifies information, displays a confirmation message, and creates an account for the new member.

**Extensions:**

3a. User enters an email address that already exists for another account:

1. User enters an email address that is used for another account.
2. System detects existing email address and prompts user to enter another email address.
3. User enters valid email address and continues registration.

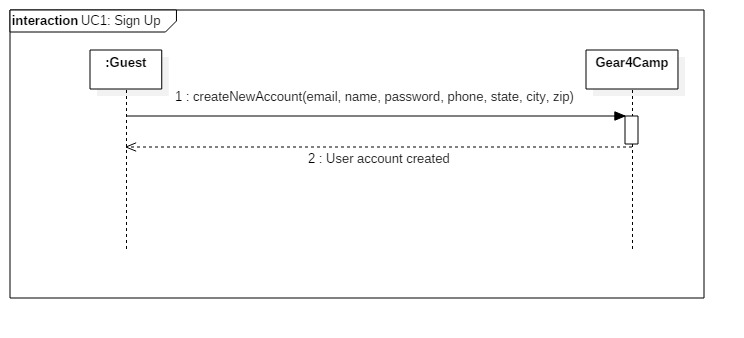
3.b. User enters an email address that is invalid:

1. User enters an email address that is not real.
2. System detects invalid email address and prompts user to enter another email address.
3. User enters valid email address and continues registration.

**Special Requirements:**

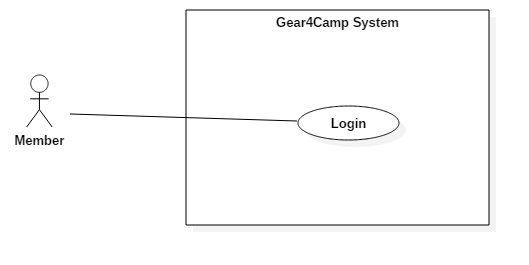
* Unique email address (to be used as username)

**Use Case Sequence Diagram:**



**Use Case UC2: Login**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: Wants to be able to log into account

**Main Success Scenario:**

1. User opens the Website and is not logged into the system yet.
2. User selects link to login.
3. System prompts the user for login information (username and password).
4. User enters his/her username and password and submits.
5. System validates the entered username and password and logs the user into the system.

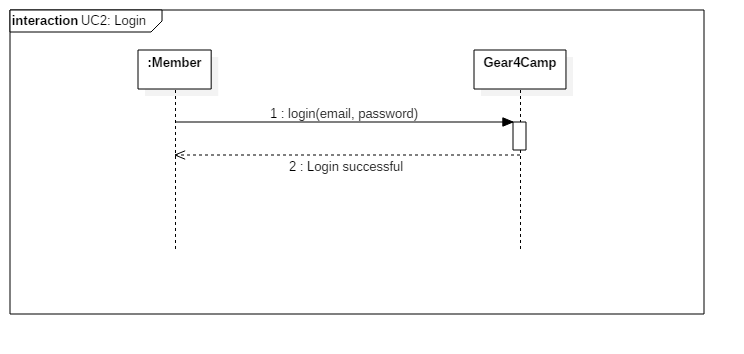
**Extensions:**

3a. User enters a wrong username or password:

1. User enters a wrong username or password (either doesn’t exist or wrong combination).
2. System detects invalid username or password and prompts user to enter another username and password combination.
3. User enters valid username and password combination and continues the login process.

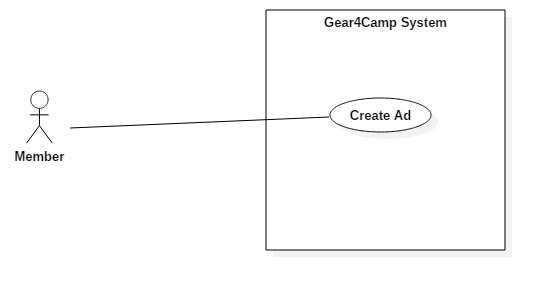
**Special Requirements:** N/A

**Use Case Sequence Diagram:**



**Use case UC3: Create AD**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: wants to post an advertisement on Gear4Camp website.

**Main Success Scenario:**

1. User opens the website.
2. User selects link to create a new ad.
3. System prompts user for ad information such as the item name, cost per day to rent, start date, end date, city, and zip code.
4. User enters ad information.
5. System saves the ad information, creates the new ad, and displays a success message.

**Extensions:**

1a. User selects link to create a new ad without being logged in.

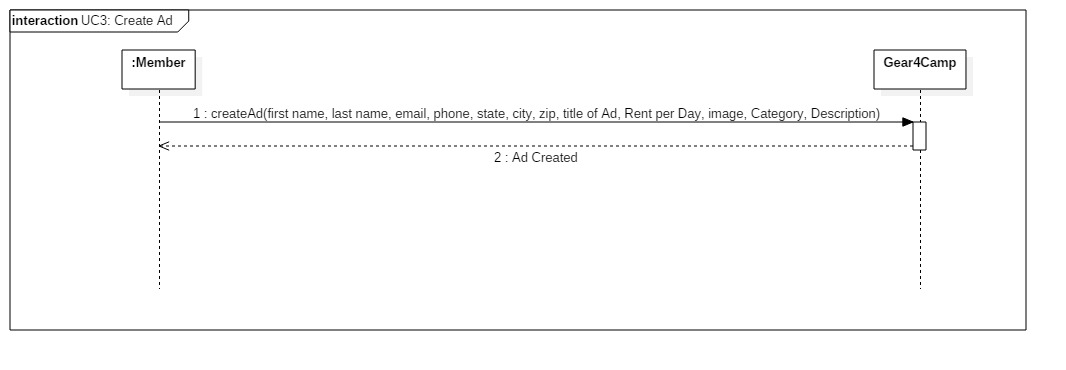
1. User selects link to create a new ad without being logged in.
2. System detects the user isn’t logged in and redirects the user to the login page.
3. User logs in (UC2: Login) and continues to the ad creation page.

3a. User enters an invalid zip code.

1. User enters an invalid zip code.
2. System detects the invalid zip code and prompts the user to enter another zip code.
3. User enters a valid zip code and continues entering ad information.

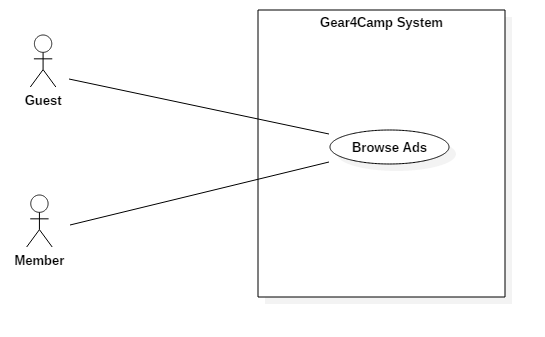
**Special Requirements:** N/A

**Use Case Sequence Diagram:**



**Use case UC4: Browse Ads**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: wants to browse ads for equipment rental.

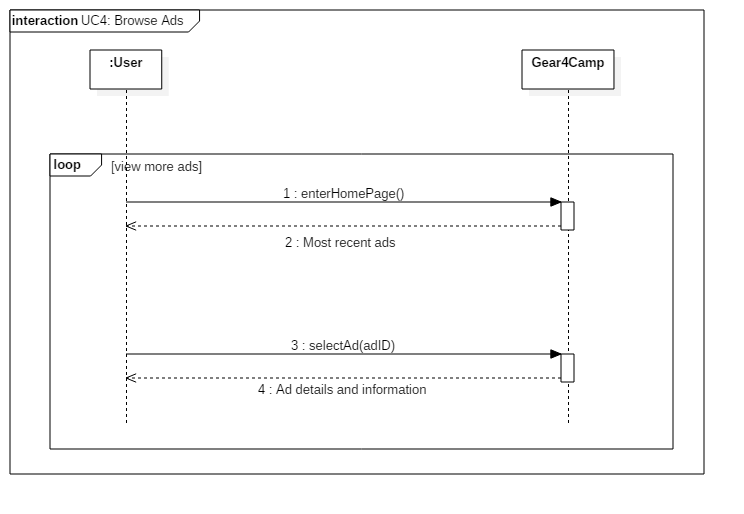
**Main Success Scenario:**

1. User opens the website.
2. User visits the home page of Gear4Camp.
3. System displays all of the latest ads available in small boxes showing summarized information including the item name, general location, and cost per day to rent.
4. User selects an ad to view in more detail.
5. System displays the selected ad in a new page with more details.
6. User repeats steps 2-5 until user is done browsing ads or decides to rent equipment.

**Extensions:** N/A

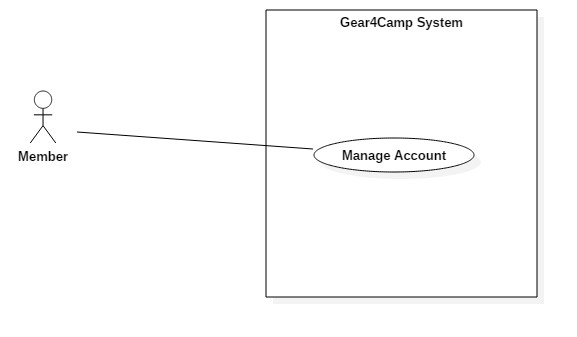
**Special Requirements:** N/A

**Use Case Sequence Diagram:**



**Use case UC5: Manage Account**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: wants to be able to view ad posting history and to manage account settings and information.

**Preconditions, Success Guarantee:** User has a valid account and successfully login into the website.

**Main Success Scenario:**

1. User opens the website.
2. User clicks the My Account button in the web page.
3. System directs user into the user’s account page.
4. User edits information about his/her account.
5. System saves the updated information and shows a confirmation message to the user.

**Postcondition:** User has updated account information.

**Extensions:**

3a. User enters an invalid zip code.

1. User enters an invalid zip code.
2. System detects the invalid zip code and prompts the user to enter another zip code.
3. User enters a valid zip code and continues entering ad information.

3b. User enters an email address that already exists for another account:

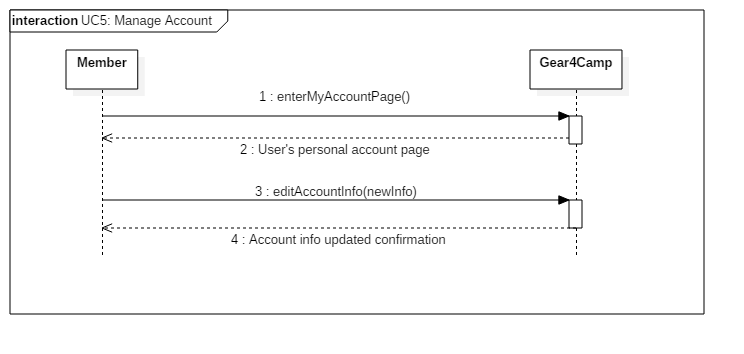
1. User enters an email address that is used for another account.
2. System detects existing email address and prompts user to enter another email address.
3. User enters valid email address and continues registration.

3.c. User enters an email address that is invalid:

1. User enters an email address that is not real.
2. System detects invalid email address and prompts user to enter another email address.
3. User enters valid email address and continues registration.

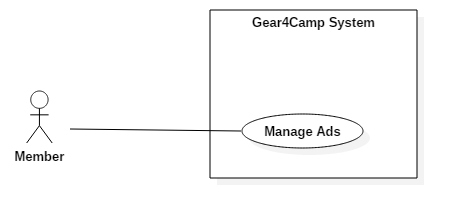
**Special Requirements:** N/A

**Use Case Sequence Diagram:**



**Use case UC6: Manage Ads**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: wants to be able to edit or delete past ad postings.

**Main Success Scenario:**

1. User selects link to manage an existing ad on his/her account page.
2. System displays current ad information such as the item name, cost per day to rent, start date, end date, city, zip code, and availability.
3. User chooses to edit / delete an ad.
4. System saves changes and displays a success message.
5. Repeat step 3-4 until user is finished editing or deleting ads.

**Extensions:**

3.a. User selects option to edit ad:

1. User selects edit ad option.
2. User updates information like item name, cost per day to rent, start date, end date, city, zip code, and availability.
3. System saves the updated ad information and displays a success message.

3a.1. User enters an invalid zip code.

1. User enters an invalid zip code.
2. System detects the invalid zip code and prompts the user to enter another zip code.
3. User enters a valid zip code and continues entering ad information.

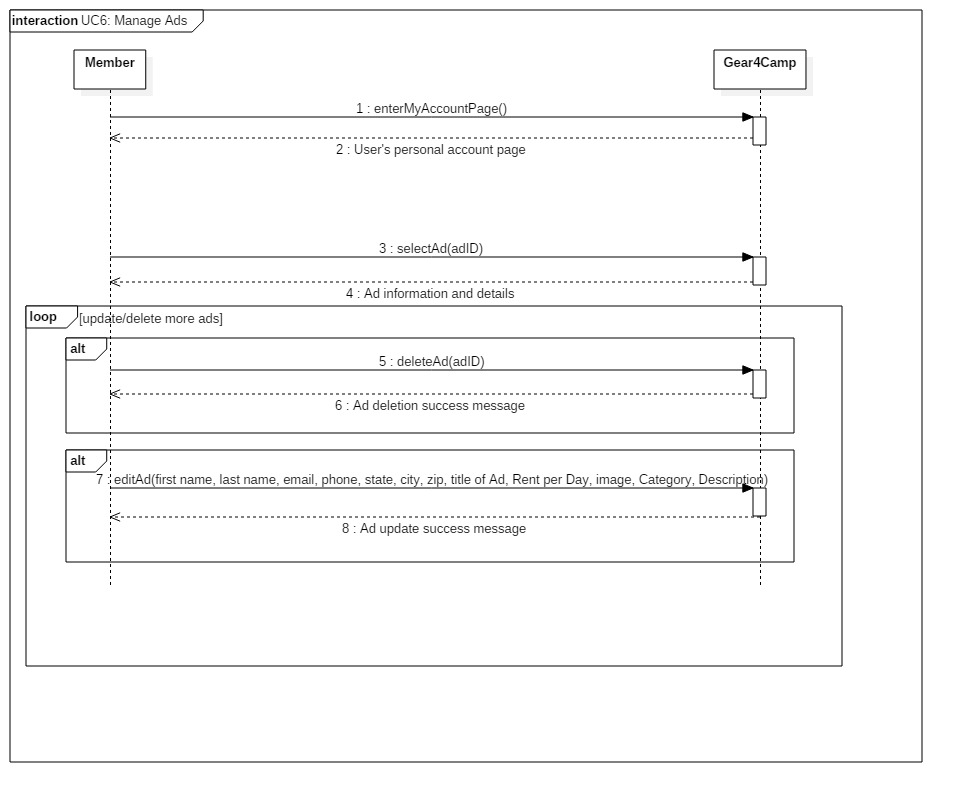
3.b. User selects option to delete ad:

1. User selects delete ad option.
2. System deletes the selected ad and displays a success message.

**Special Requirements:**

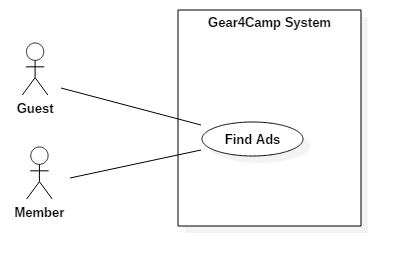
* User must have at least 1 ad posting to edit or delete a previous ad posting.

**Use Case Sequence Diagram:**



**Use case UC7: Find Ads**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary)

**Stakeholders and Interests:**

* User: wants to be able to find ads for specific equipment.

**Main Success Scenario:**

1. User opens the website.
2. User visits the home page of Gear4Camp.
3. User looks for an ad for specific equipment via using the search function, filtering ads with specific criteria, or filter ads with a category.
4. System displays ads relevant to the query.
5. User repeats steps 2-3 until user is done searching for ads.

**Extensions:**

2.a. User uses the search function to find ads:

1. User enters keywords into the search bar.
2. System displays ads from search results containing the keywords.

2.b. User filters ads based on specific criteria:

1. User filters ads based on specific criteria such as sorting from low to high, newest ads, owner ratings, etc.
2. System displays ads from filtering results based on the specified criteria.

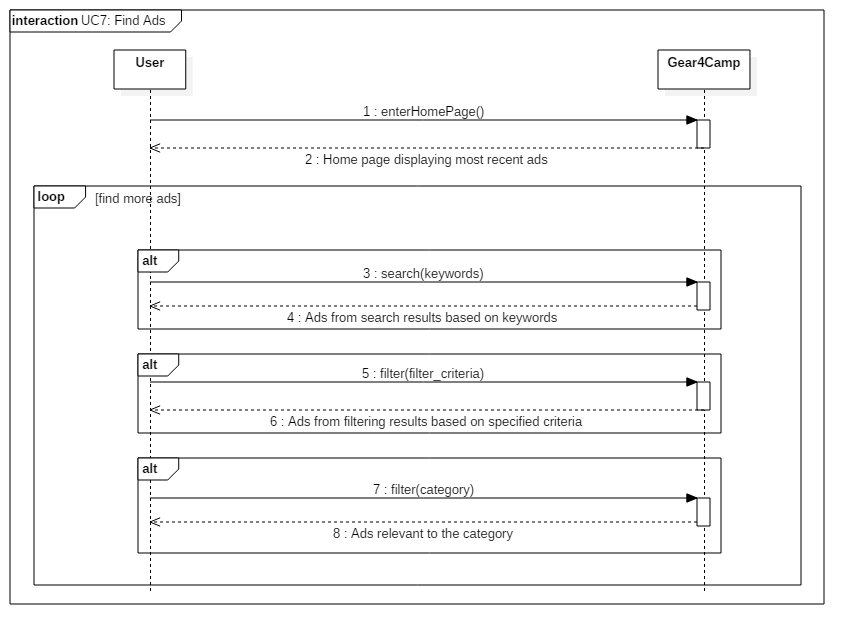
2.c. User filters ads with a category:

1. User chooses a category for ads to be shown.
2. System displays ads relevant to the category.

**Special Requirements:**

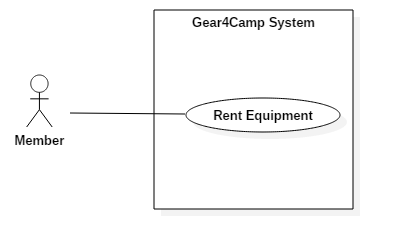
* There must be at least 1 ad created previously.

**Use Case Sequence Diagram:**



**Use case UC8: Rent Equipment**

**Use Case Diagram:**



**Scope:** Gear4Camp application

**Level:** User goal

**Actors:** User (Primary), AdPoster (Supporting)

**Stakeholders and Interests:**

* User: wants to be able to rent equipment through the website after viewing an ad.

**Main Success Scenario:**

1. User opens the website.
2. User selects an ad of interest.
3. System displays ad in new page with more details.
4. User selects the option to send a request to the ad poster to rent the item.
5. System forwards the request to the ad poster.
6. System displays request sent successful message to user.
7. User contacts the ad poster via email or phone number to get more information and negotiate transaction details.
8. Ad poster responds to the user’s message and the negotiation terms are complete.
9. Ad poster accepts the request through the website, which automatically sets the specified ad for the item to unavailable.
10. System confirms the user’s request to rent message as accepted.
11. User repeats steps 1-9 until user is done trying to rent equipment.

**Extensions:**

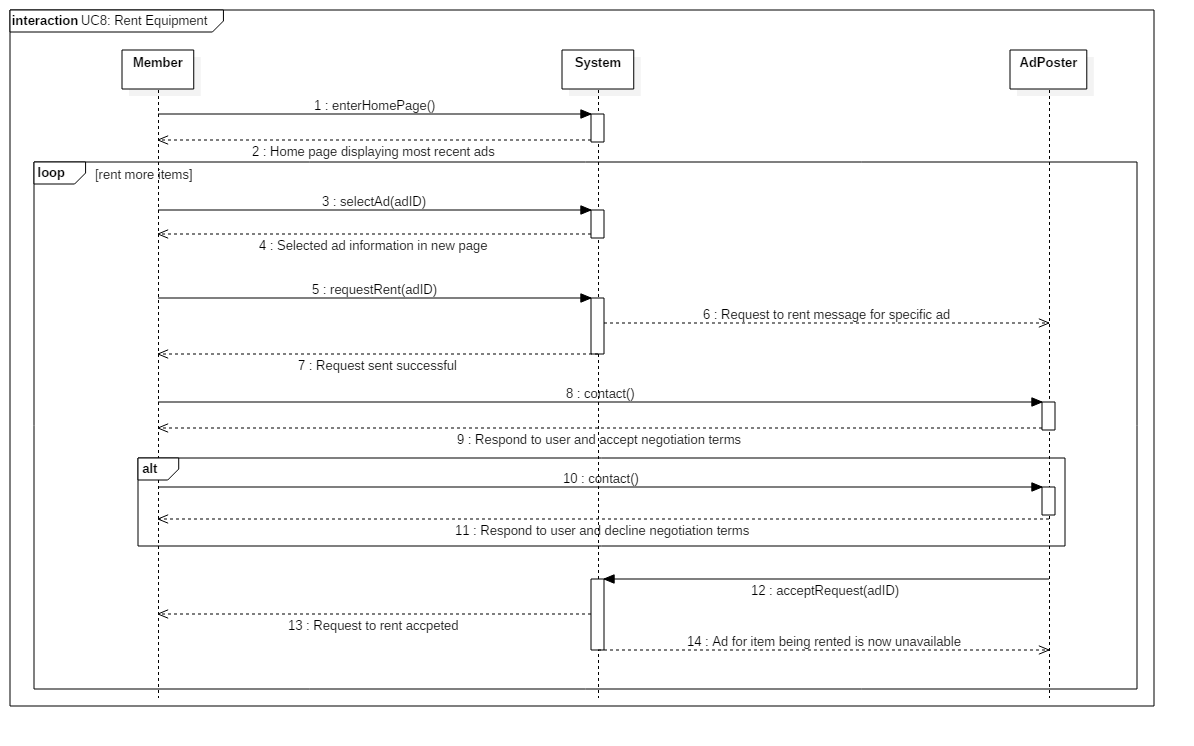
7.a. Ad poster declines request to rent out the item:

1. System confirms the user’s request to rent message as declined.
2. User repeats steps 1-9 until user is done trying to rent equipment.

**Special Requirements:**

* There must be at least 1 ad created previously.

**Use Case Sequence Diagram:**



**11. Operation Contracts**

**Contract CO1: createNewAccount**

**Operation:** createNewAccount(email, name, password, dob)

**Cross Reference:** UC1: Sign Up

**Preconditions:** None

**Postconditions:**

* An Account instance acc was created
* acc was associated with the current user
* Attributes of acc were initialized
* ID\_number attribute of acc assigned a unique number

**Contract CO2: login**

**Operation:** login(email, password)

**Cross Reference:** UC2: Login

**Preconditions:** User has registered for an account (UC1).

**Postconditions:**

* User is logged in successfully and redirected to home page.

**Contract CO3: createAd**

**Operation:** createAd(item, cost\_per\_day, start\_date, end\_date, city, zip\_code, picture)

**Cross Reference:** UC3: Create Ad

**Preconditions:** User is logged into account (UC2).

**Postconditions:**

* An Ad instance, ad was created
* ad was associated with acc
* adAD attribute of ad assigned a unique number

**Contract CO4: selectAd**

**Operation:** selectAd(adID)

**Cross Reference:** UC4: Browse Ad

**Preconditions:** At least 1 ad must exist and is available (UC3).

**Postconditions:**

* ad was associated with the current user

**Contract CO5: editAccount**

**Operation:** editAccount(email, name, password, dob)

**Cross Reference:** UC5: Manage Account

**Preconditions:** User is logged in (UC2).

**Postconditions:**

* acc.email may have been updated
* acc.name may have been updated
* acc.password may have been updated
* acc.dob may have been updated

**Contract CO6: editAd**

**Operation:** editAd(item, cost\_per\_day, start\_date, end\_date, city, zip\_code, picture)

**Cross Reference:** UC6: Manage Ad

**Preconditions:** User is logged in (UC2) and has created at least 1 ad (UC3).

**Postconditions:**

* ad.item may have been updated
* ad.cost\_per\_day may have been updated
* ad.start\_date may have been updated
* d.end\_date may have been updated
* ad.city may have been updated
* ad.zip\_code may have been updated
* ad.picture may have been updated

**Contract CO7: deleteAd**

**Operation:** deleteAd(adID)

**Cross Reference:** UC6: Manage Ad

**Preconditions:** User is logged in (UC2) and has created at least 1 ad (UC3).

**Postconditions:**

* ad was deleted

**Contract CO8: searchAd**

**Operation:** searchAd(keywords)

**Cross Reference:** UC7: Find Ad

**Preconditions:** At least 1 ad must exist and is available (UC3).

**Postconditions:**

* Multiple Ad instances were associated (displayed) with the current user

**Contract CO9: filterAd**

**Operation:** filterAd(categories, criteria)

**Cross Reference:** UC7: Find Ad

**Preconditions:** At least 1 ad must exist and is available (UC3).

**Postconditions:**

* Multiple Ad instances were associated (displayed) with the current user

**Contract CO10: rentAd**

**Operation:** rentAd(owner\_id, renter\_id, ad\_id)

**Cross Reference:** UC8: Rent Ad

**Preconditions:** At least 1 ad must exist and is available (UC3).

**Postconditions:**

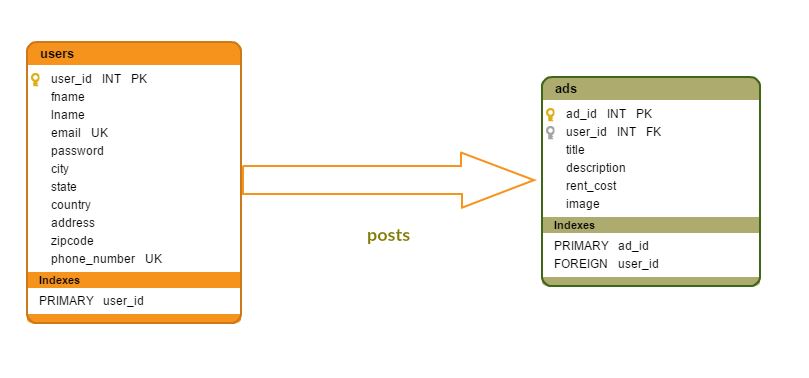
* A RentContract instance rent\_contract was created.
* rent\_contract was associated with the current user
* rent\_contract was associated with ad
* rent\_contract was associated with ad owner

**12. Database Schema**

For Iteration 2 we have 2 tables in our database.

1. Users: This database has the details of all the users like First name, Last name, Email ID, phone number, address and more
2. Ads: This database has details of all the ads which are present right now. For each ad the details such as Title, description, image, cost and many more.

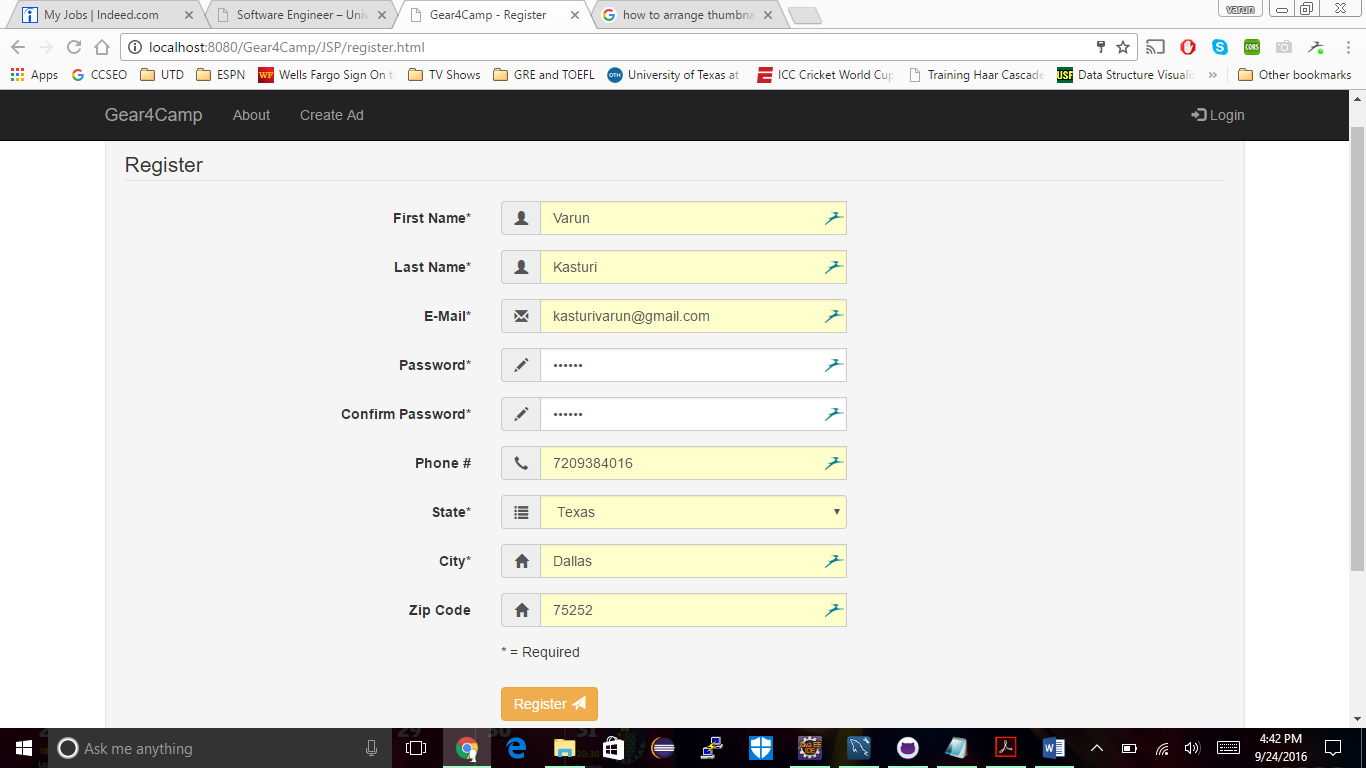
When user posts an ad it will stored in the database.



**13. Testing**

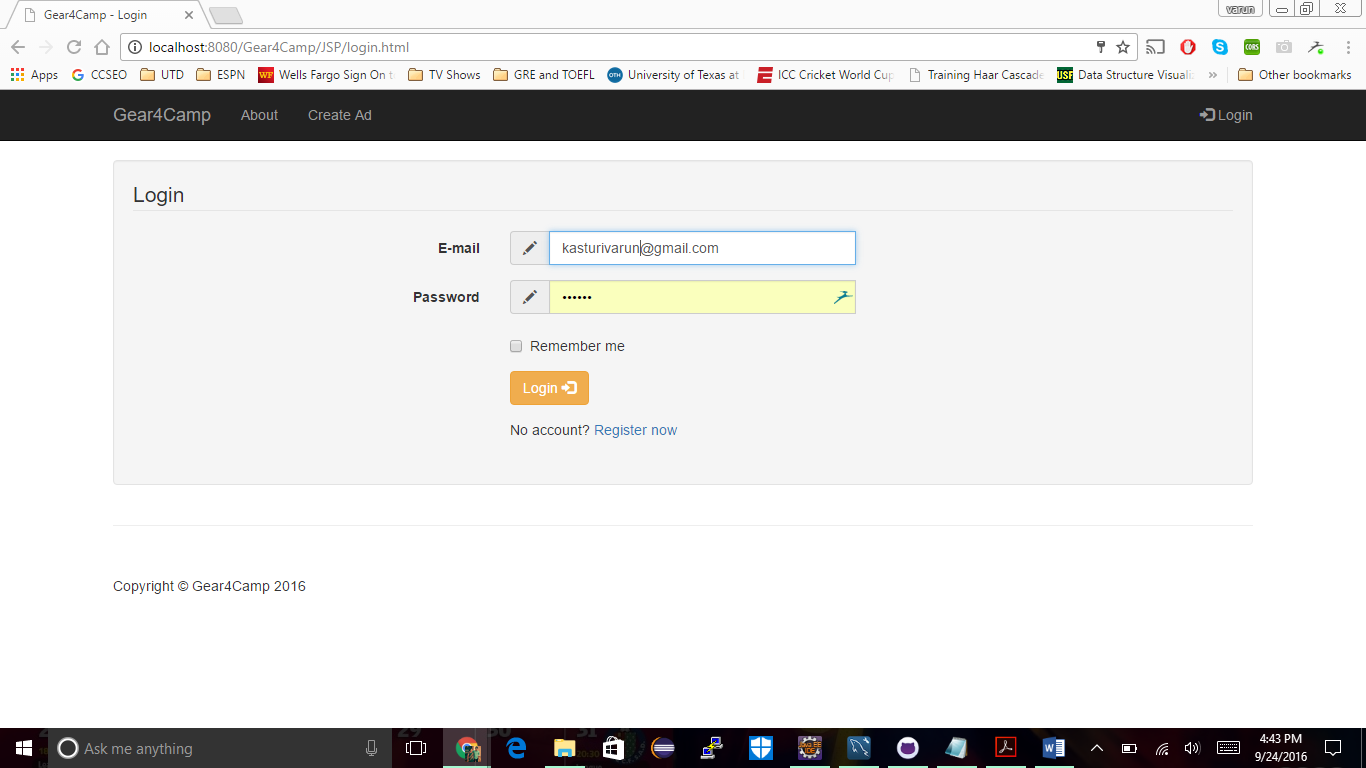
**Testing UC1: Sign Up**

We submitted test information into the register form. The information was successfully stored in the database and the account was created.



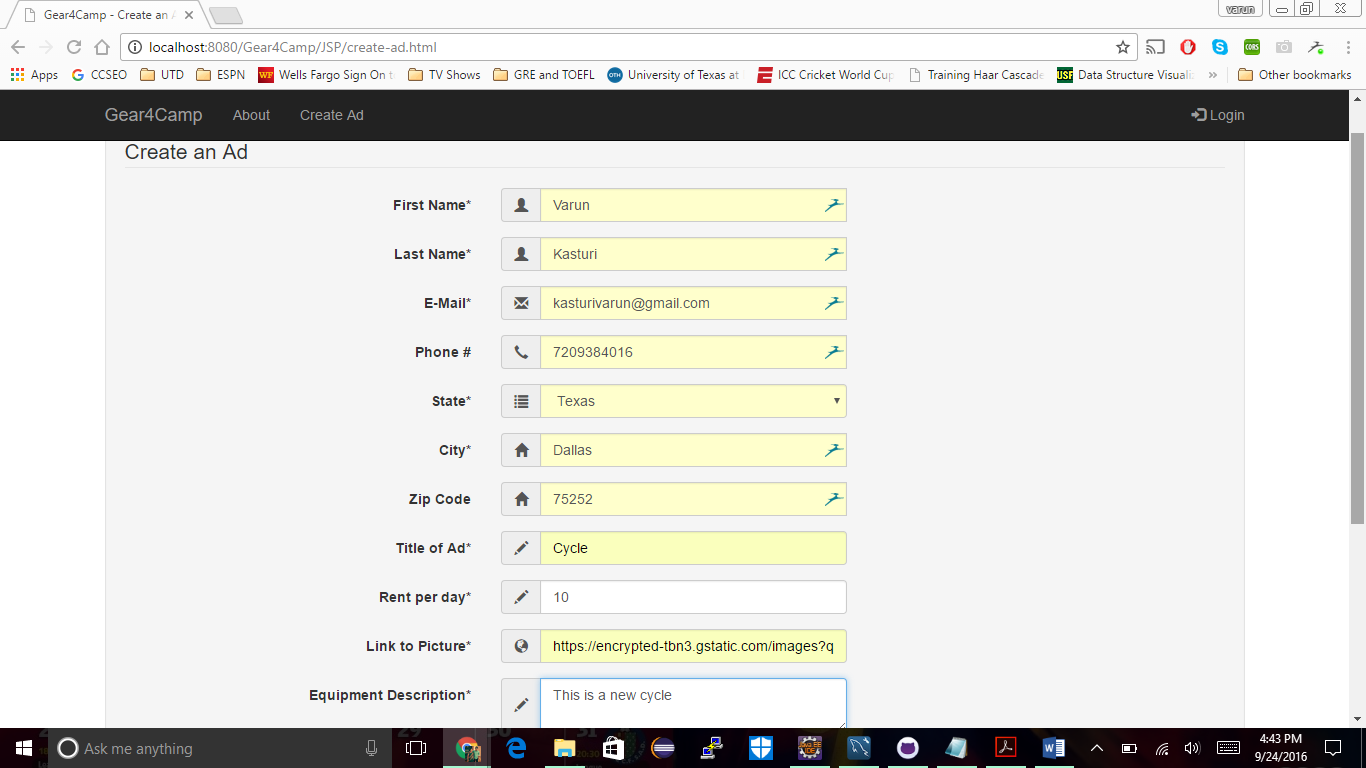
**Testing UC2: Login**

We tested logging into the dummy account made previously. The application was able to check and verify the submitted login details with the database and login was successful.



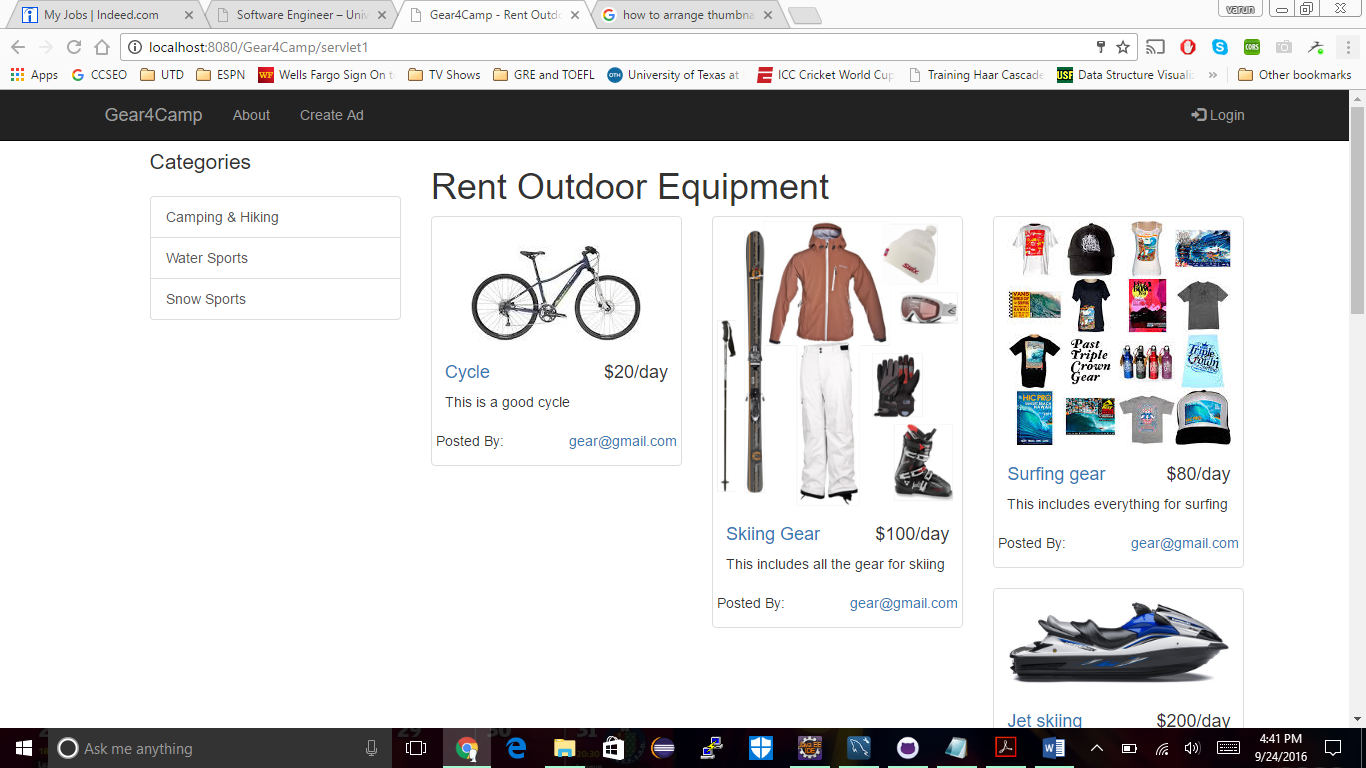
**Testing UC3: Create Ad**

After logging in, we submitted information to create an ad. The data was stored successfully on the database and the ad was created successfully.



**Testing UC4: Display Ads**

After creating an ad, the application pulled the ad information from the database and displayed on the homepage successfully.



**14. Project Management**

**Expectations for Iteration 3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **ID** | **Task** | **Assigned** | **Dependency** | **Comments** |
| 10/20/16 | 3.1 | Use cases for posting information without login | Varun, Sneha |  |  |
| 10/20/16 | 3.2 | UML Diagrams for posting information without login | Varun, Sneha |  |  |
| 10/20/16 | 3.3 | Use case for user rating system | Kainan, Henry |  | Including payment for rent and security deposit |
| 10/20/16 | 3.4 | UML Diagrams for user rating system | Kainan, Henry |  |  |
| 10/21//16 | 3.5 | Implementation for posting without login | All | 1.1 |  |
| 10/21/16 | 3.6 | Implementation for user rating system | All | 1.3 |  |
| 11/10/16 | 3.7 | UI Implementation and testing | All |  |  |
| 11/10/16 | 3.8 | Database Layer implementation and testing | All |  |  |
| 11/15/16 | 3.9 | Milestone: Iteration 3 | All |  | Members must rate each other both ways after transactions. Members will be able to favorite ads they are interested in. |