

# Sprint 1 Demo

Joanne, KD, Christian, Kayla

# System Requirements and Functions

1. Creating Members and Groups
2. Members Joining Groups
3. Creating Posts

## Testing

1. Unit Tests
2. Integration Tests

# Create Members

- As a **Member** I want to participate in organizing and contributing to Q&A communities.
- Create an instance of **Member**
  - **Requirements:**
    - a. **first/last name**
    - b. **email address**
    - c. **user name**
      - unique from email address
    - d. **time stamp**

# Create Groups

- **Groups** organize posts and allow members to interact with them.
- Create an instance of `Group`
  - **Requirements:**
    - a. title
    - b. description
    - c. time stamp

# Join Members to Groups

- Members acquire a membership to a group.

1. Create an instance of Membership

- **Dependencies:**

- a. instance of Member

- b. instance of Group

## Join Members to Groups Cont.

2. Use the `Member.joinGroup` method

- **Dependencies:**

- a. instance of `Member` , `Group` , and `Membership`
- b. member has valid group-membership

- **Requirements:**

- a. group
- b. time stamp

# Create Posts

- Members create posts in their groups.
- Create an instance of `Answer` or `Question`.
  - **Dependencies:**
    - a. instances of `Group`, `Member`, and `Membership`
    - b. member has valid group-membership
  - **Requirements:**
    - a. content
    - b. timestamp

# Create Questions

- Members create posts in the form of a question.
- Create an instance of `Question`.
  - **Dependencies:**
    - a. instances of `Group`, `Member`, and `Membership`
    - b. member has valid group-membership
  - **Requirements:**
    - a. title
    - b. content
    - c. timestamp



# Create Answers

- Members create posts in the form of an answer.
- Create an instance of `Answer`.
  - **Dependencies:**
    - a. instances of `Group`, `Member`, `Membership`, and `Question`
    - b. member has valid group-membership
    - c. question has valid group-membership
  - **Requirements:**
    - a. content
    - b. timestamp

# Unit Tests

- Unit Tests assed the functionality of individual class behaviors.

# Integration Tests

- Integration Tests assess the functionality of integrated class behaviors.