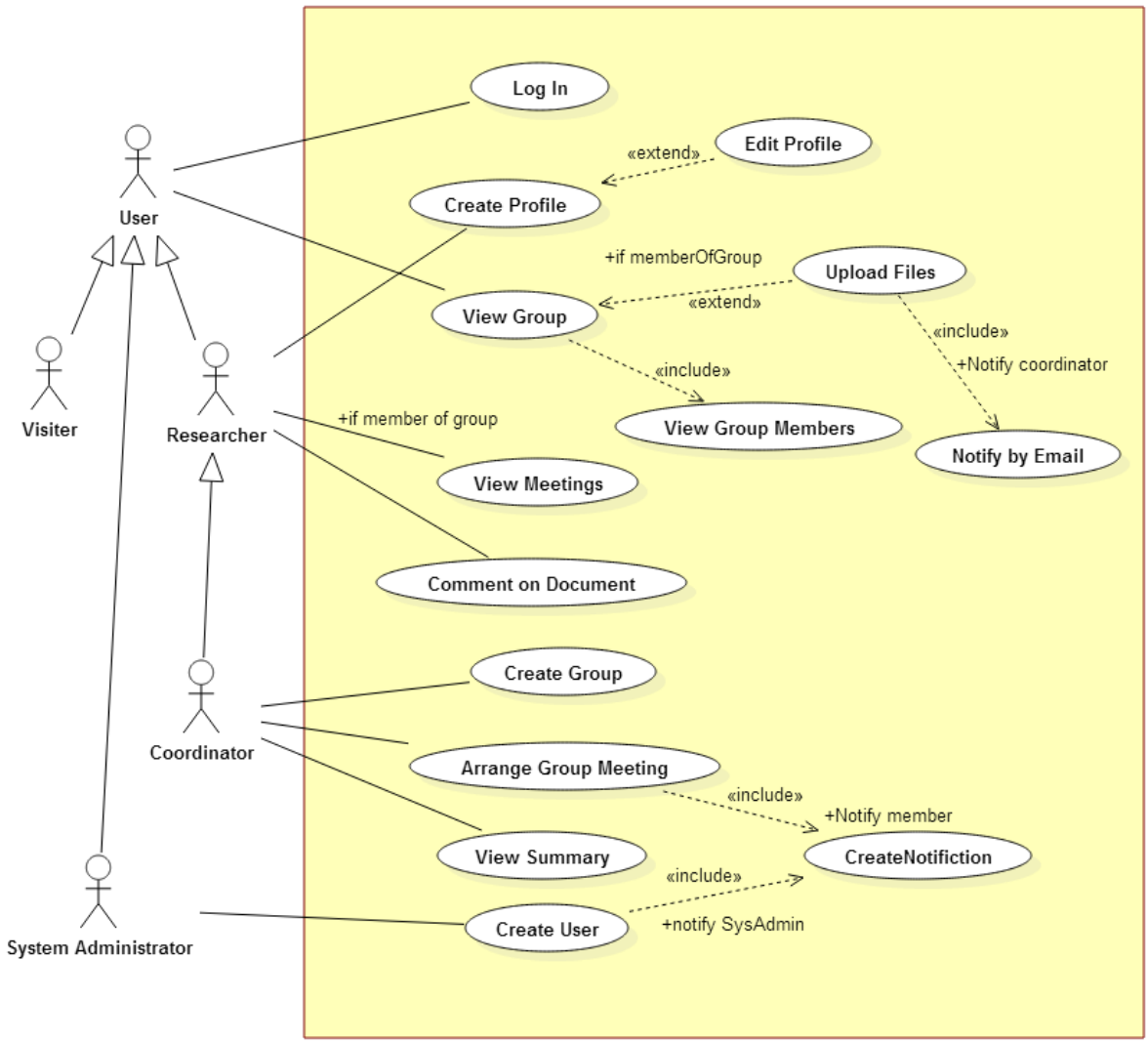
# SENG2050 – Assignment 3 (RGMS)

Requirements Specification

## Use Case Diagram



# Requirements Analysis

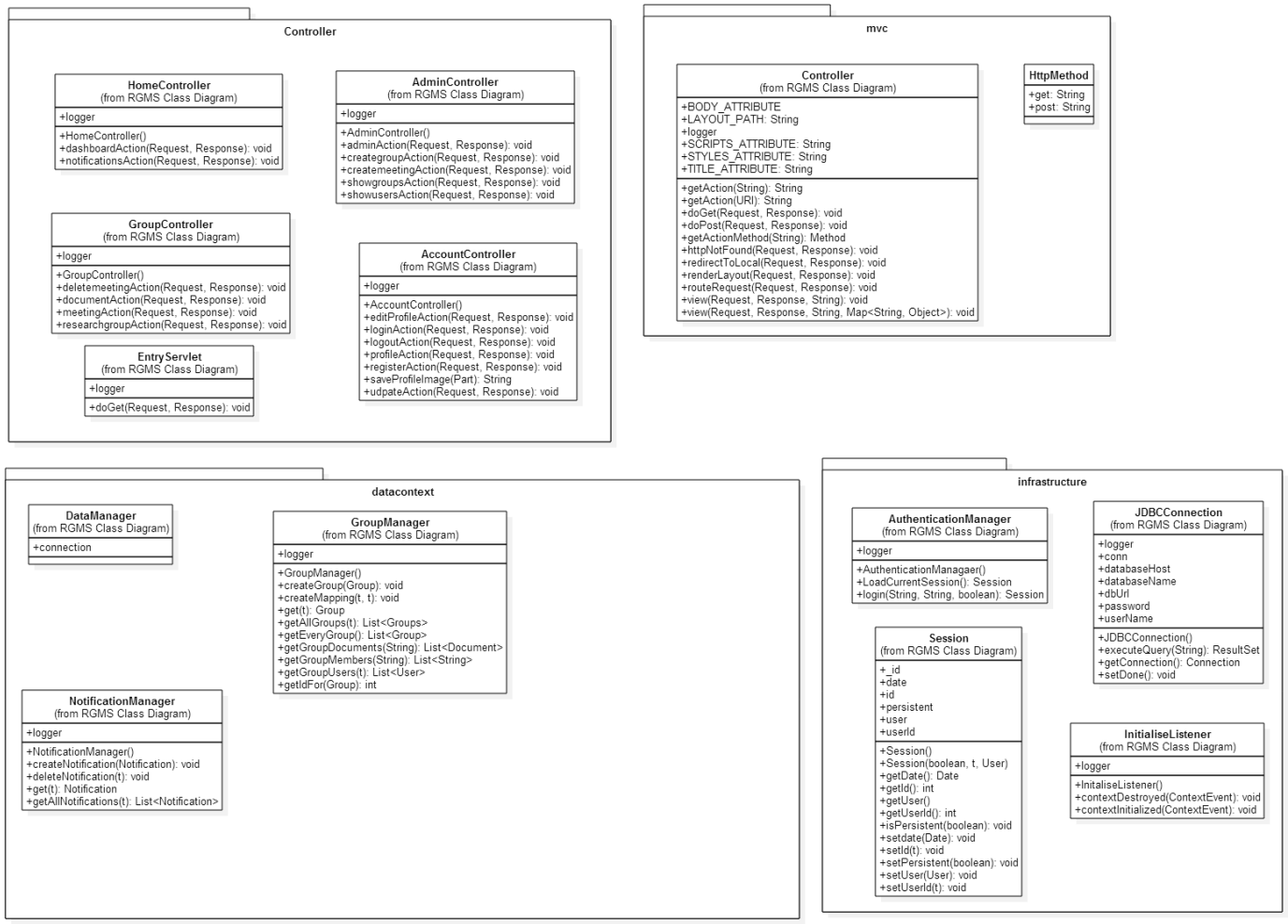
## In Scope

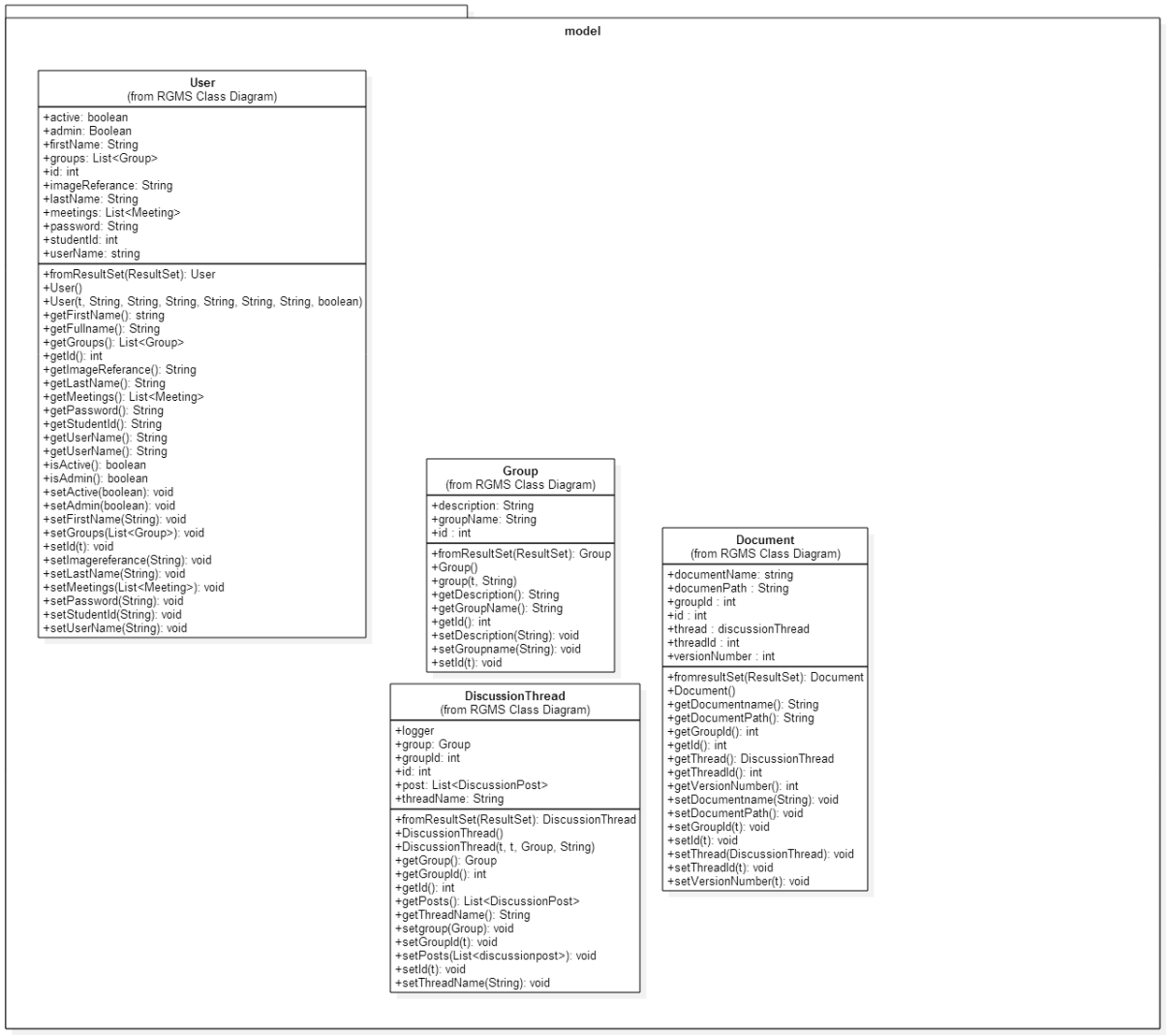
- Users should be able to log into the system using their username and password
- Researchers can participate in multiple groups
- Groups have one coordinator
- Coordinators act as system administrators for the system
- Coordinators enrol users in the system
- Coordinators set up the research groups
- Coordinators can create meetings
- Users of groups are notified when meetings are created
- Group participants can upload documents that are only visible to members of that group
- Uploaded documents have date and version control
- Coordinators can see access records of documents
- Users are notified when documents are uploaded
- Users can post comments on documents
- Users can edit their profile information
- Group members can create threads in the group that are only visible to members of that group
- Users can see who is in their group and view other member's profiles
- Notifications will be created within the system

## Out of Scope

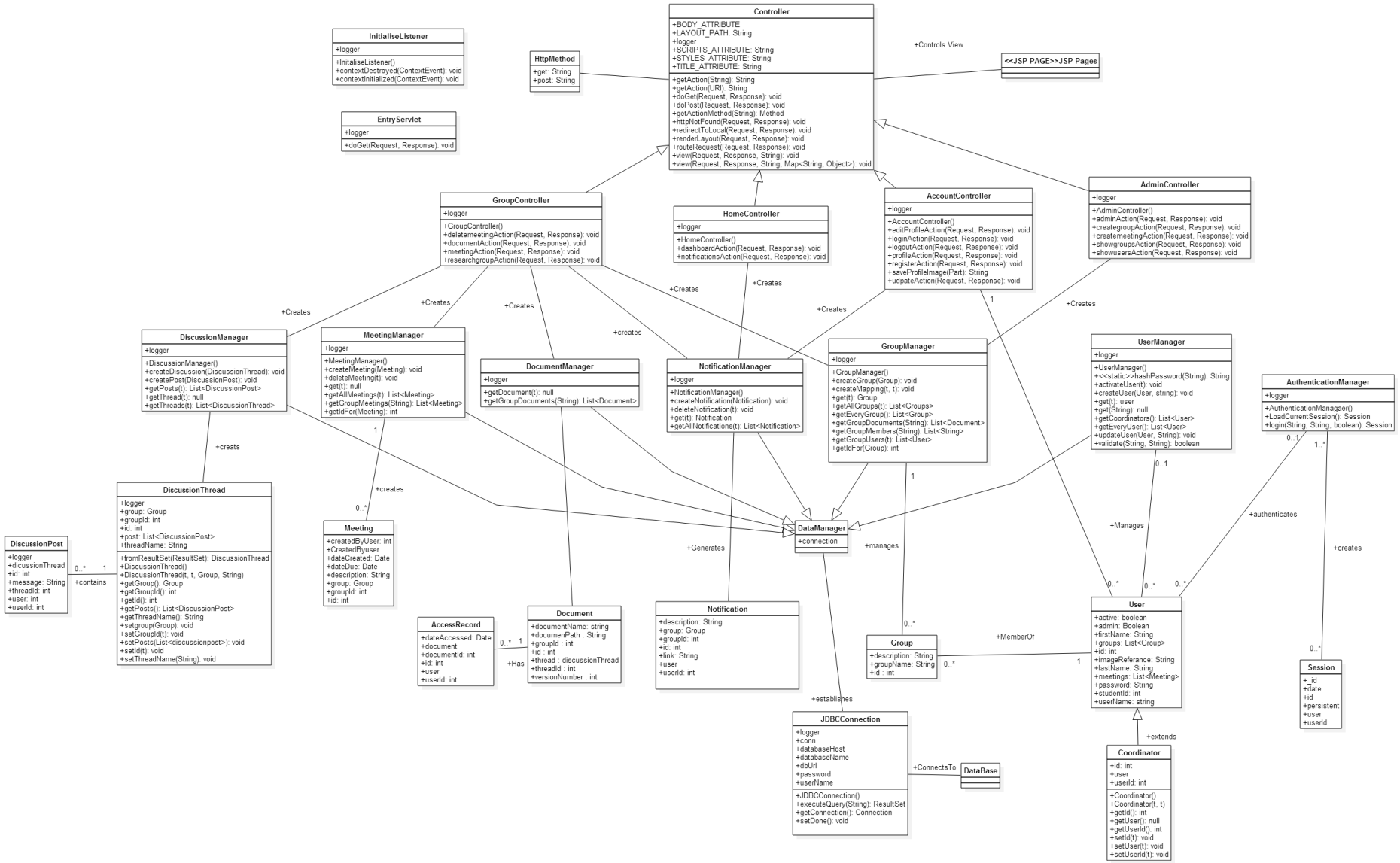
- Coordinators cannot leave their Groups
- Meetings do not have an associated discussion thread
- Users cannot decline from meetings
- System admins only activate user accounts. They do not create users
- Email will not be used to send notifications

# Package Diagram



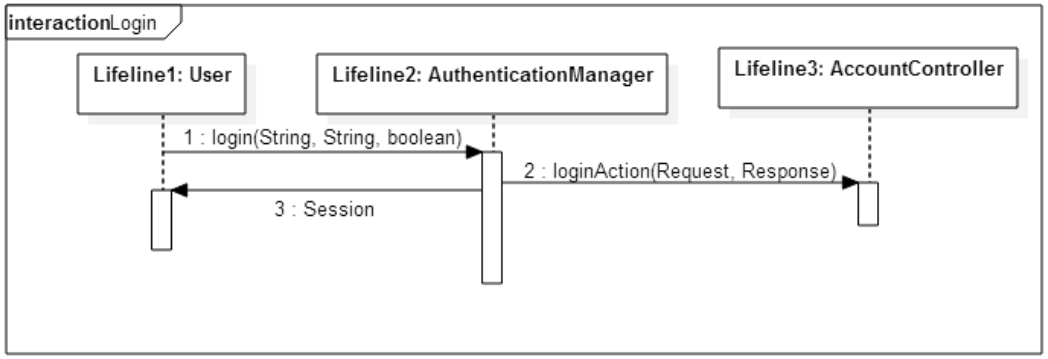

model
User
(from RGMS Class Diagram)
+active: boolean
+admin: Boolean
+firstName: String
+groups: List<Group>
+id: int
+imageReferance: String
+lastName: String
+meetings: List<Meeting>
+password: String
+studentId: int
+userName: string
+fromResultSet(ResultSet): User
+User()
+User(t, String, String, String, String, String, String, boolean)
+getFirstName(): string
+getFullname(): String
+getGroups(): List<Group>
+getId(): int
+getImageReferance(): String
+getLastName(): String
+getMeetings(): List<Meeting>
+getPassword(): String
+getStudentId(): String
+getUserName(): String
+getUserName(): String
+isActive(): boolean
+isAdmin(): boolean
+setActive(boolean): void
+setAdmin(boolean): void
+setFirstName(String): void
+setGroups(List<Group>): void
+setId(t): void
+setImagereferance(String): void
+setLastName(String): void
+setMeetings(List<Meeting>): void
+setPassword(String): void
+setStudentId(String): void
+setUserName(String): void
Group
(from RGMS Class Diagram)
+description: String
+groupName: String
+id : int
+fromResultSet(ResultSet): Group
+Group()
+group(t, String)
+getDescription(): String
+getGroupName(): String
+getId(): int
+setDescription(String): void
+setGroupname(String): void
+setId(t): void
Document
(from RGMS Class Diagram)
+documentName: string
+documenPath : String
+groupId : int
+id : int
+thread : discussionThread
+threadId : int
+versionNumber : int
+fromresultSet(ResultSet): Document
+Document()
+getDocumentname(): String
+getDocumentPath(): String
+getGroupId(): int
+getId(): int
+getThread(): DiscussionThread
+getThreadId(): int
+getVersionNumber(): int
+setDocumentname(String): void
+setDocumentPath(): void
+setGroupId(t): void
+setId(t): void
+setThread(DiscussionThread): void
+setThreadId(t): void
+setVersionNumber(t): void
DiscussionThread
(from RGMS Class Diagram)
+logger
+group: Group
+groupId: int
+id: int
+post: List<DiscussionPost>
+threadName: String
+fromResultSet(ResultSet): DiscussionThread
+DiscussionThread()
+DiscussionThread(t, t, Group, String)
+getGroup(): Group
+getGroupId(): int
+getId(): int
+getPosts(): List<DiscussionPost>
+getThreadName(): String
+setgroup(Group): void
+setGroupId(t): void
+setPosts(List<discussionpost>): void
+setId(t): void
+setThreadName(String): void

# Class Diagram

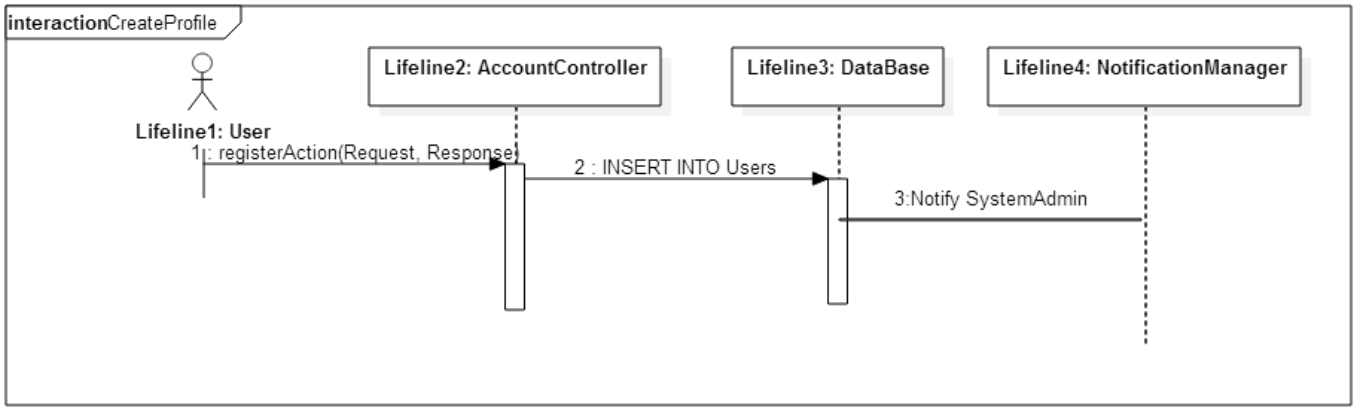


# Sequence Diagrams

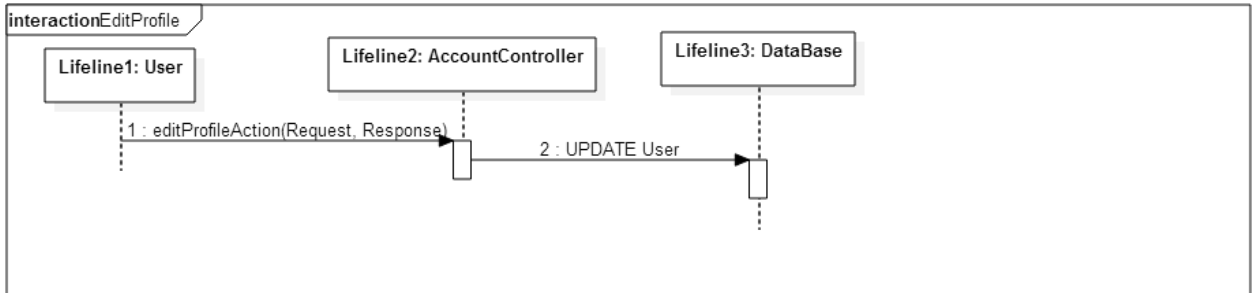
## Login



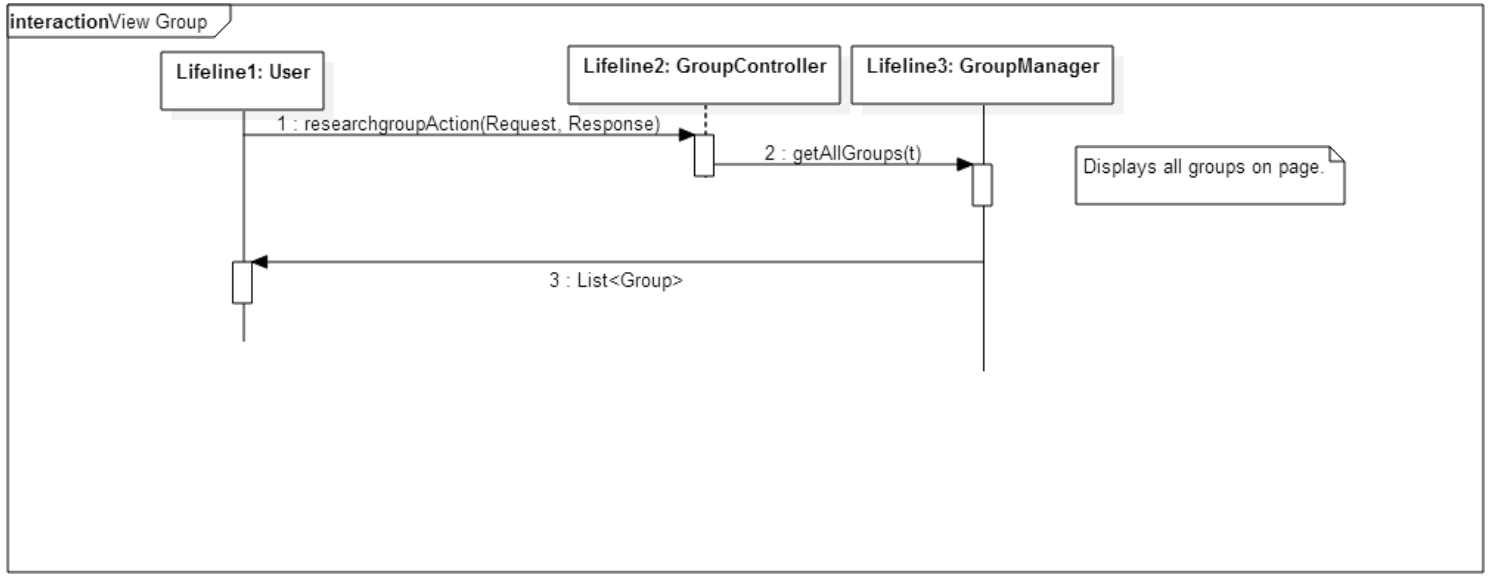

## Create Profile



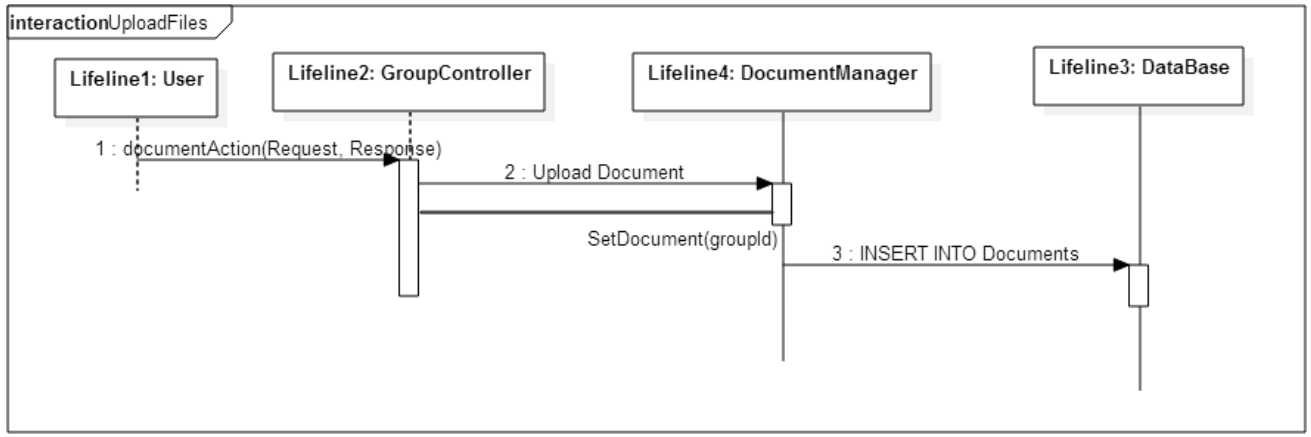
## Edit Profile



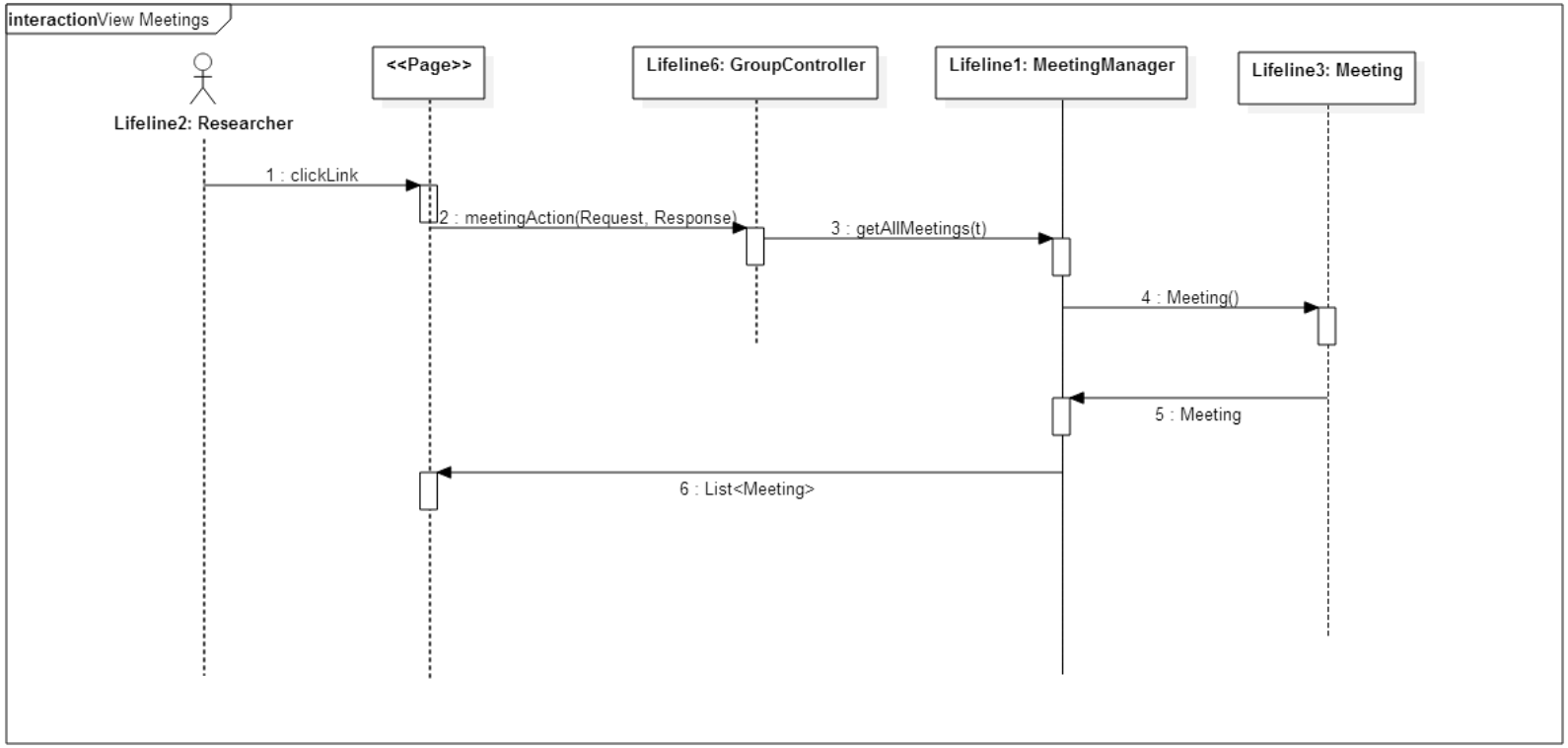

## View Group



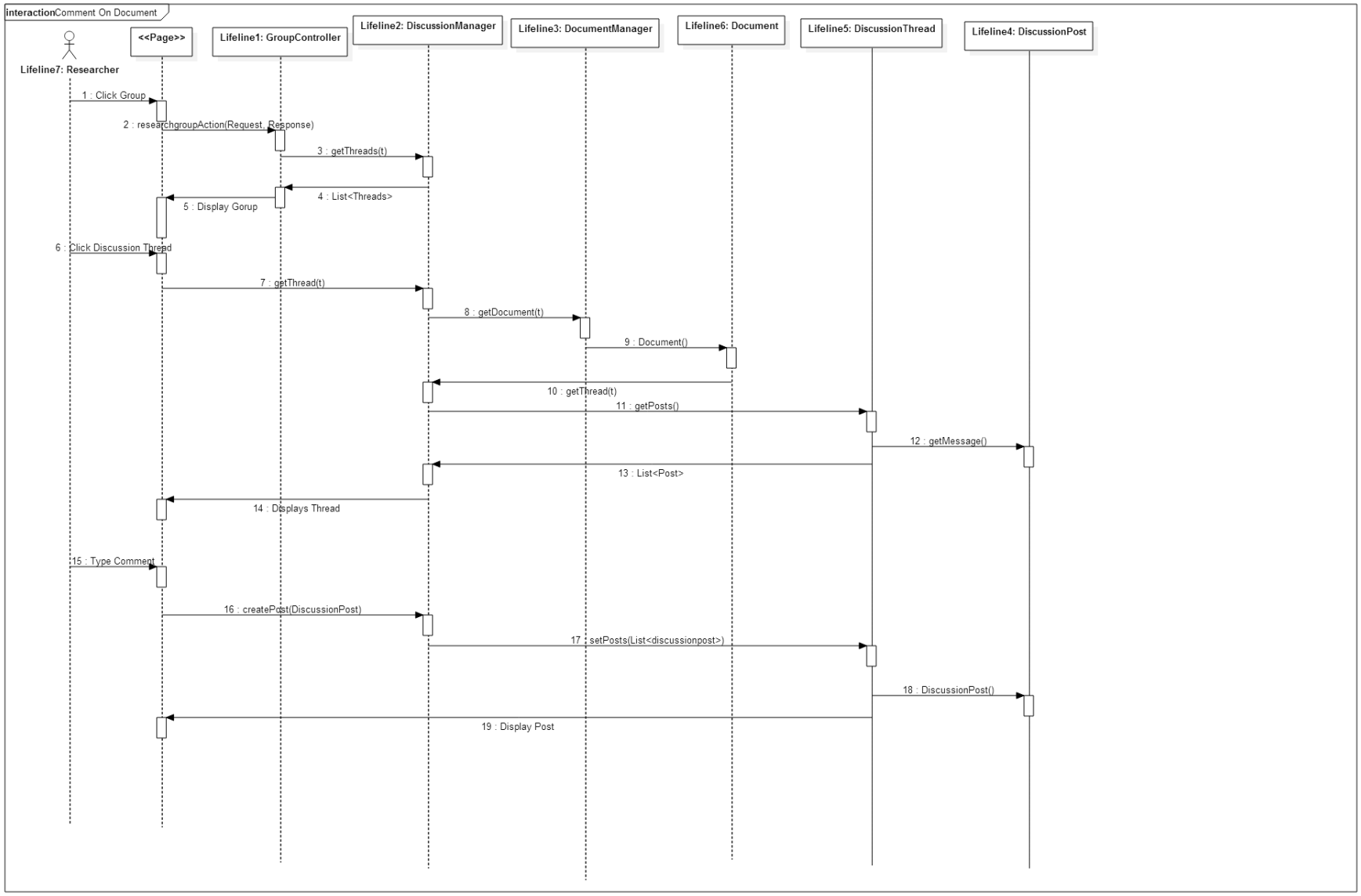
## Upload Files



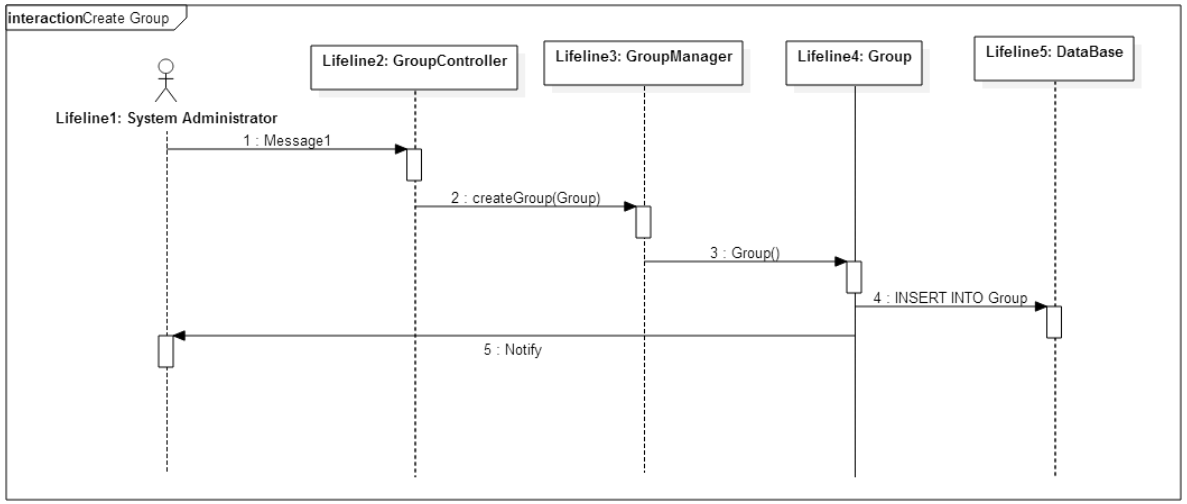
## View Meetings



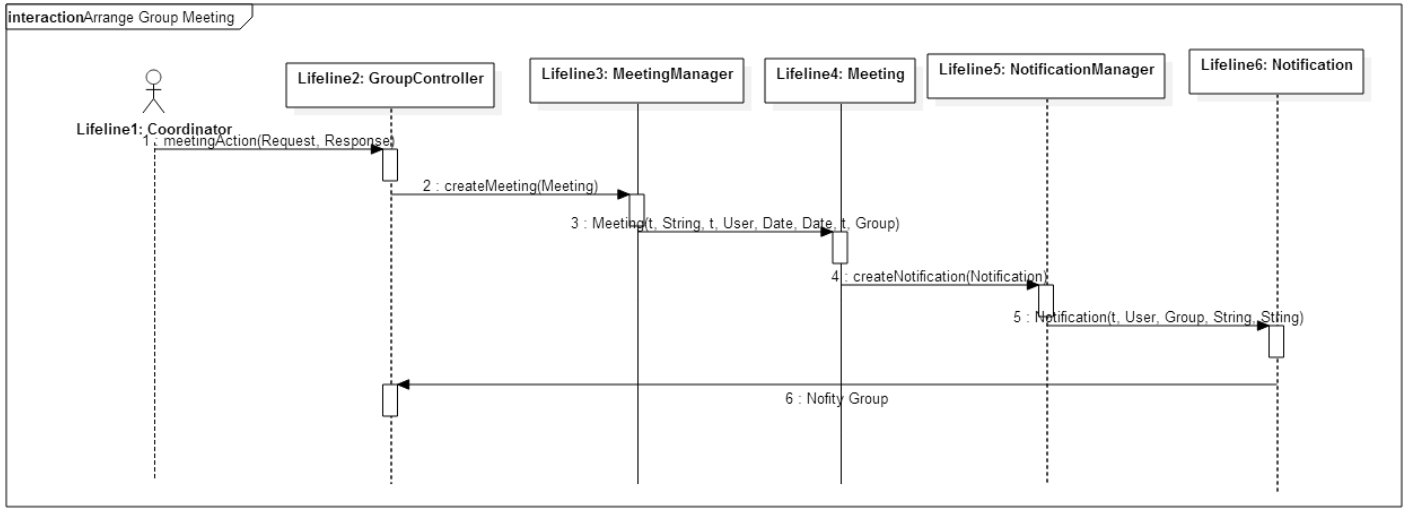
## Comment on Document



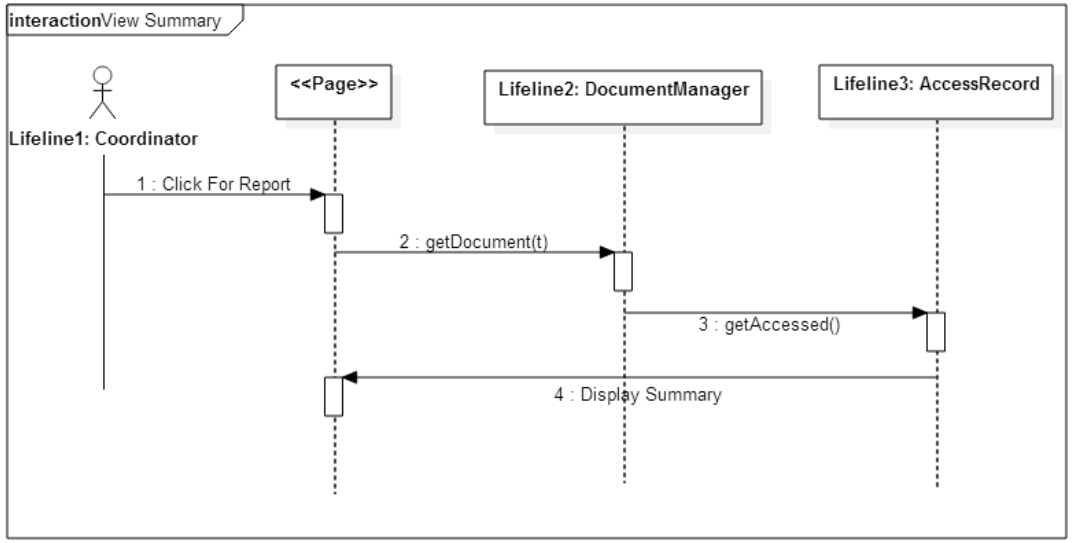
## Create Group



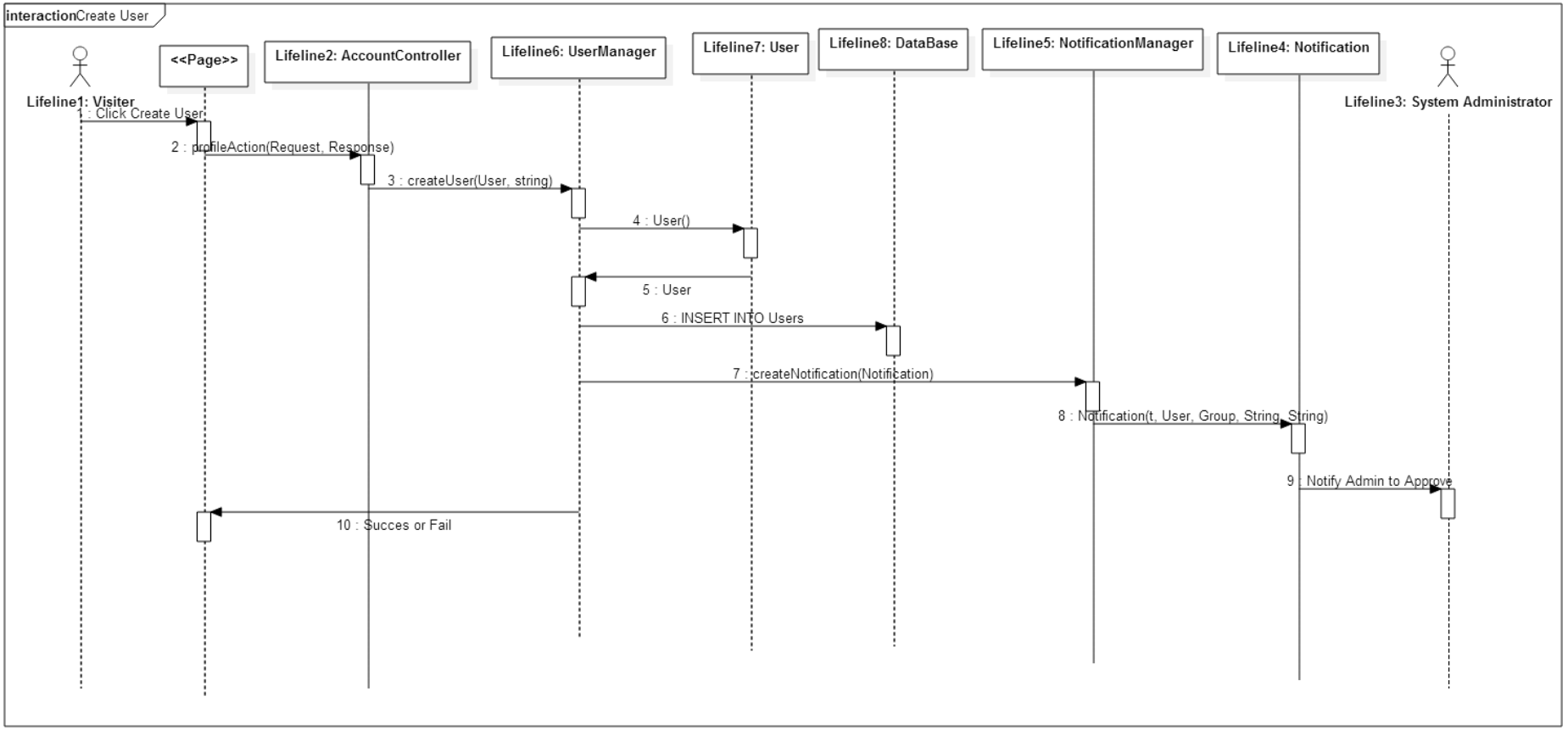

## Arrange Group Meeting



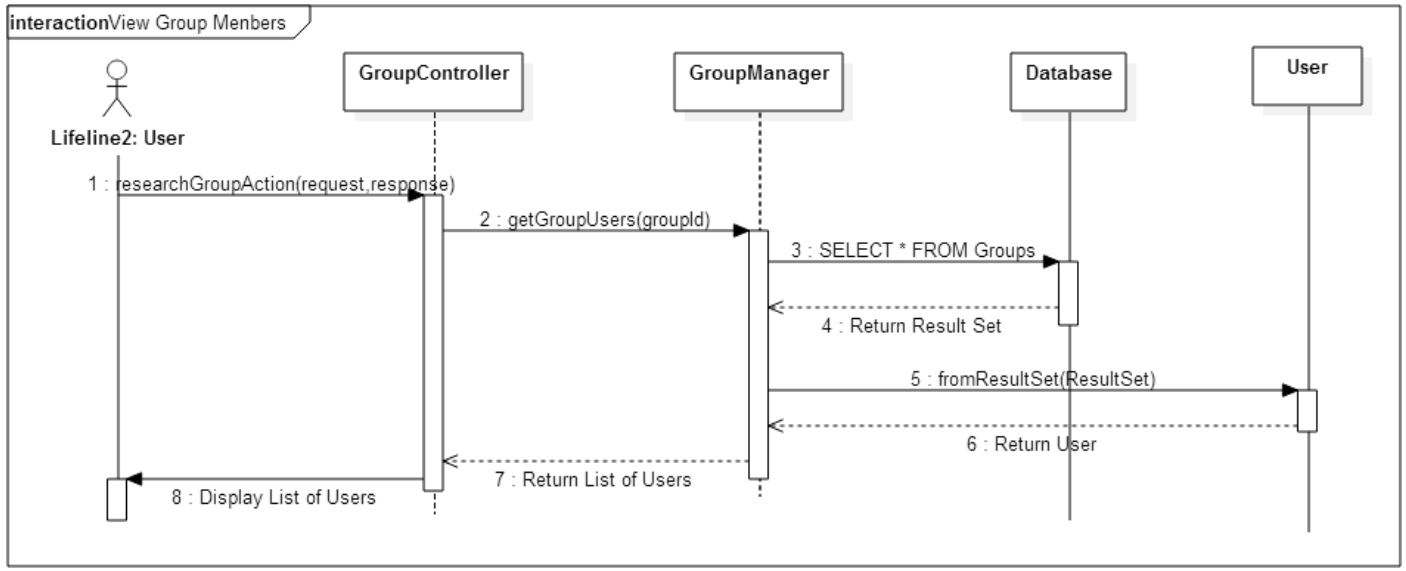
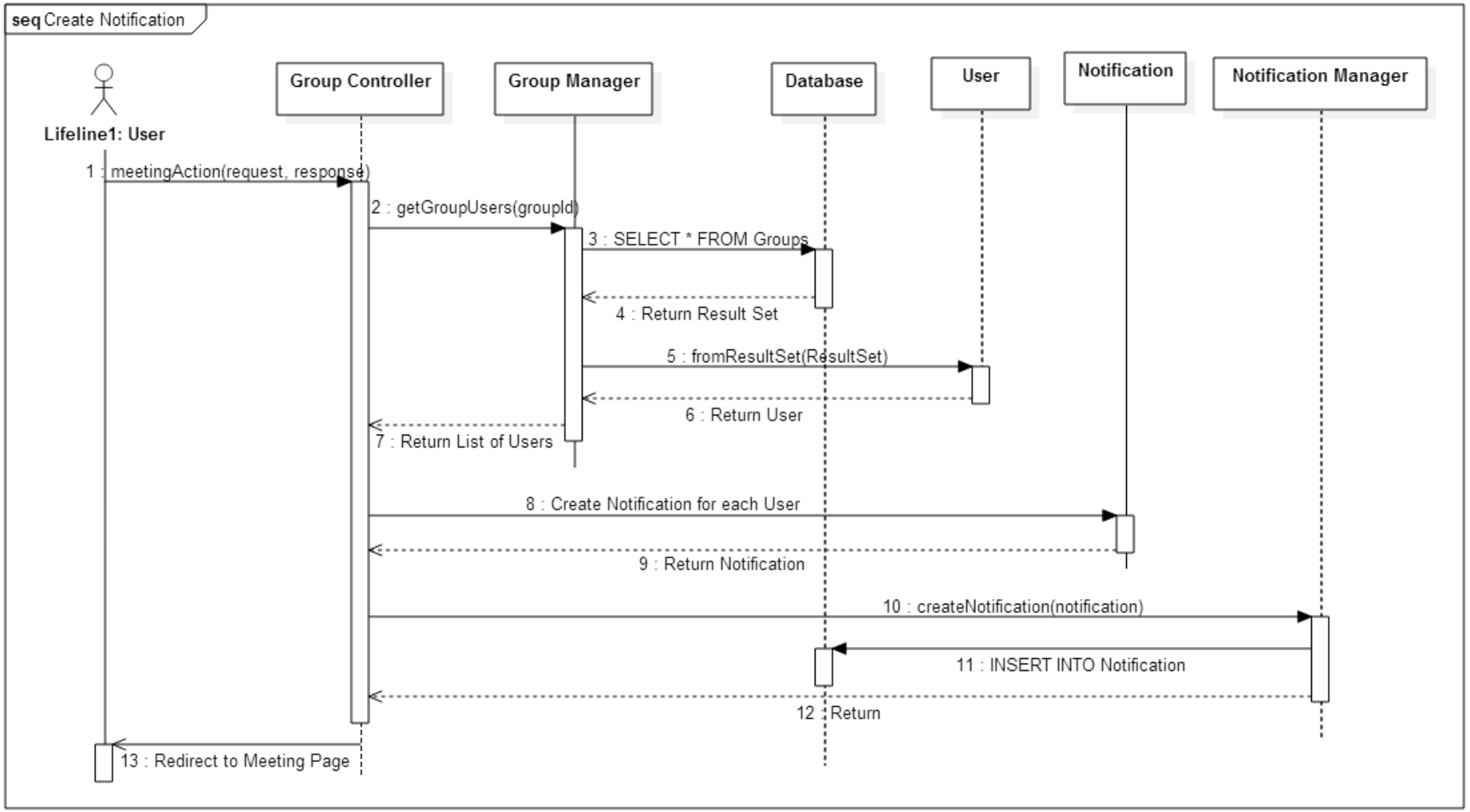
## View Summary



## Create User


interactionCreate User
Lifeline1: Visiter
<<Page>>
Lifeline2: AccountController
Lifeline6: UserManager
Lifeline7: User
Lifeline8: DataBase
Lifeline5: NotificationManager
Lifeline4: Notification
Lifeline3: System Administrator
1 : Click Create User
2 : profileAction(Request, Response)
3 : createUser(User, string)
4 : User()
5 : User
6 : INSERT INTO Users
7 : createNotification(Notification)
8 : Notification(t, User, Group, String, String)
9 : Notify Admin to Approve
10 : Succes or Fail

## View Group Members



## Create Notification