

The LinkedIn System

LinkedIn is an employment-focused Social Network platform to manage the user's professional identity. It helps them find new opportunities, grow their careers, & connect with the people they know & trust professionally.

Find the best employment profile page represents their education, experience, skills, achievements & recommendations. Create a post regarding a topic of interest, comment on a post, invite other users to become a connection, message other users, & create groups to form a community on a post.

Expectation from the Interviewee

- How can users search the job opportunities for themselves?
- How are companies able to find out the perfect match for their Job Opening?
- How are users able to search other user's profile?
- Can users search for other users using their company name/ employment history & job type?
- How are users able to connect with other users?
- On a page connect with other pages as will
- How can users follows or unfollow the pages without blocking a direct connection
- Can the users create both groups & pages in addition to this user profile

- Can both the users & companies create groups & pages?
- Can the individual user post a Job
- Can a user join any group, or are there any criteria for joining the group?
- How will users be notified of preselcted events?
- How much control do users have in choosing what notifications they can receive

Requirement Collection

- R1: Users should be able to add information to their profile including education, experience, achievements & skills.
- R2: Users should be able to search for, & also view, pages, groups & other users.
- R3: Users should be able to send & cancel notification requests. They should also be able to respond to the connection requests of other users by either accepting or ignoring them.
- R4: Users should be able to follow users without adding them as their connection.
- R5: Users should be able to view their number of connections, profile views, post impressions & search appearances.
- R6: Users should be able to request & give recommendations to other users.

R7: Users should be able to write a new posts.

R8: Users should be able to react, share & comment on a post.

They should be able to react or comments on an existing comments.

A user should be able to send & receive messages from

R9: A user should be able to create company pages. Users other user.

The users should be able to follow other company pages.

R10: Should be able to follow other company pages.

The system should send a notification to the user to inform

R11: them about messages, connection request, or comments on their post.

Company pages should have a list of Job Openings the user can apply.

R12: Company pages should have a list of Job Openings the user can apply.

A user should be able to create & join groups.

R13: A user should be able to create & join groups.

Actors

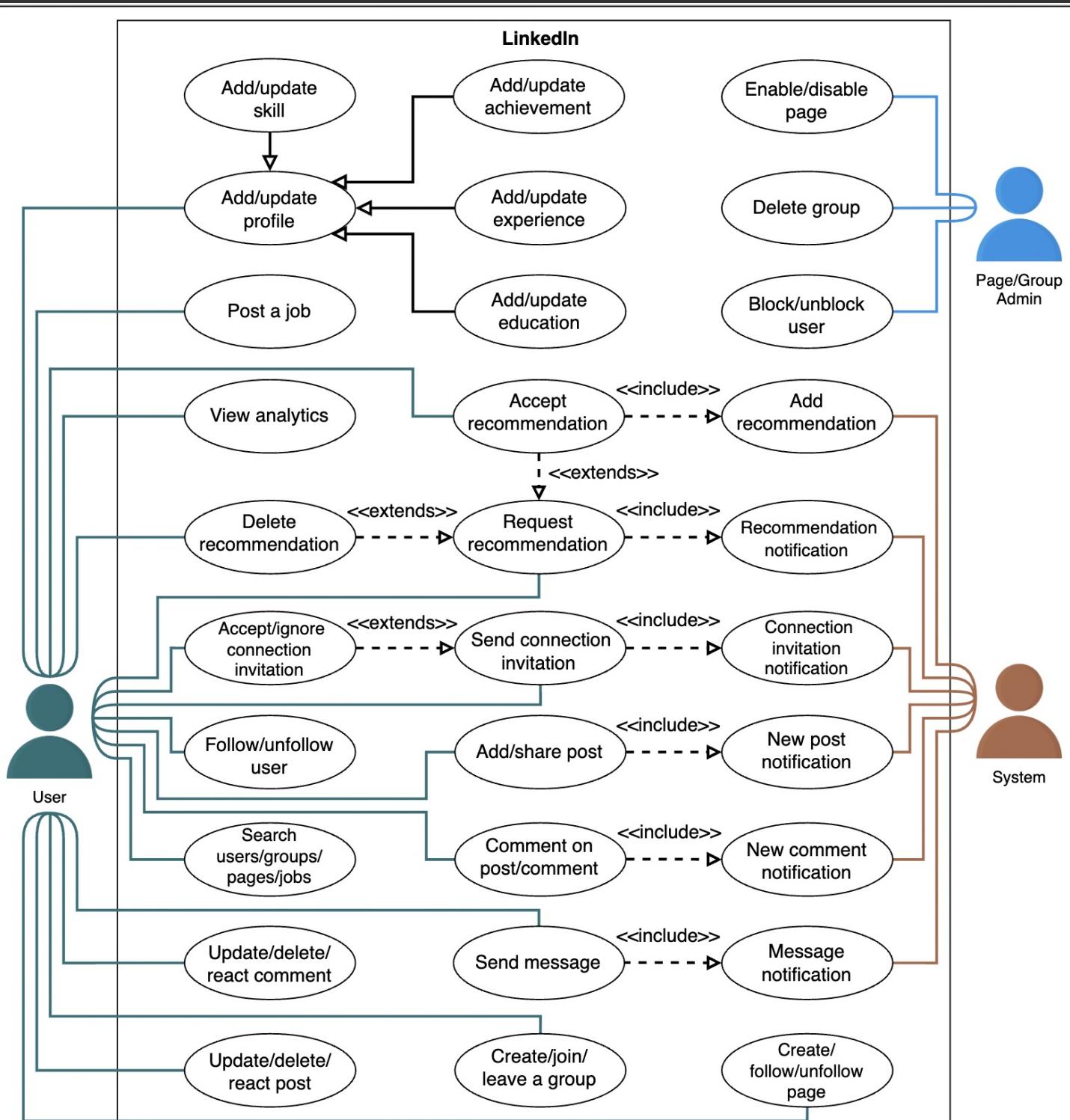
Primary Actors

- User
- Admin

Secondary Actors

- System

User	Admin	System
Add/update profile	Block/unblock user	New post notification
Follow/unfollow user	Enable/disable page	New comment notification
Send message	Delete group	Connection invitation notification
Send connection invitation		Message notification
Accept/ignore connection invitation		Recommendation notification
Create/follow/unfollow page		Add recommendation
Create/join/leave a group		
Add/share post		
Update/delete/react post		
Comment on post/comment		
Update/delete/react comment		
Search users/groups/pages/jobs		
Post a job		
View analytics		
Request recommendation		
Accept recommendation		
Delete recommendation		

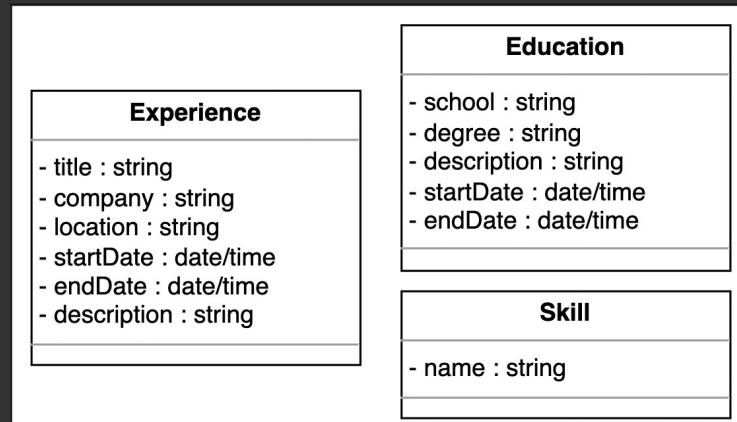


Class Diagram

1. Account

Account
- status : AccountStatus
- accountId : string
- username : string
- password : string
- email : string
+ resetPassword() : bool

2. Education, Experience, Skill



3. Recommendations, Achievements & Analytics

Recommendation
- userId : int
- createdOn : date/time
- description : string
- isAccepted : bool

Achievement
- title : string
- dateAwarded : date/time
- description : string

Analytics
- searchAppearances : int
- profileViews : int
- postImpressions : int
- totalConnections : int

4. Profile

Profile
- profilePicture : byte {list}
- coverPhoto : byte {list}
- headline : string
- about : string
- gender : string
- experiences : Experience {list}
- education : Education {list}
- skills : Skill {list}
- achievements : Achievement {list}
- recommendations : Recommendation {list}
- analytics : Analytics
+ addExperience() : bool
+ addEducation() : bool
+ addSkill() : bool
+ addAchievement() : bool

5. CompanyPage, Job & Group

CompanyPage	Job	Group
<ul style="list-style-type: none"> - pageId : int - name : string - description : string - type : string - companySize : int - createdBy : User - jobs : Job {list} <ul style="list-style-type: none"> + createJobPosting() : bool + deleteJobPosting(Job) : bool 	<ul style="list-style-type: none"> - jobId : int - jobTitle : string - dateOfPosting : date/time - description : string - company : CompanyPage - employmentType : string - location : Address - status : JobStatus 	<ul style="list-style-type: none"> - groupId : int - name : string - description : string - totalMembers : int - members : User {list} <ul style="list-style-type: none"> + updateDescription() : bool

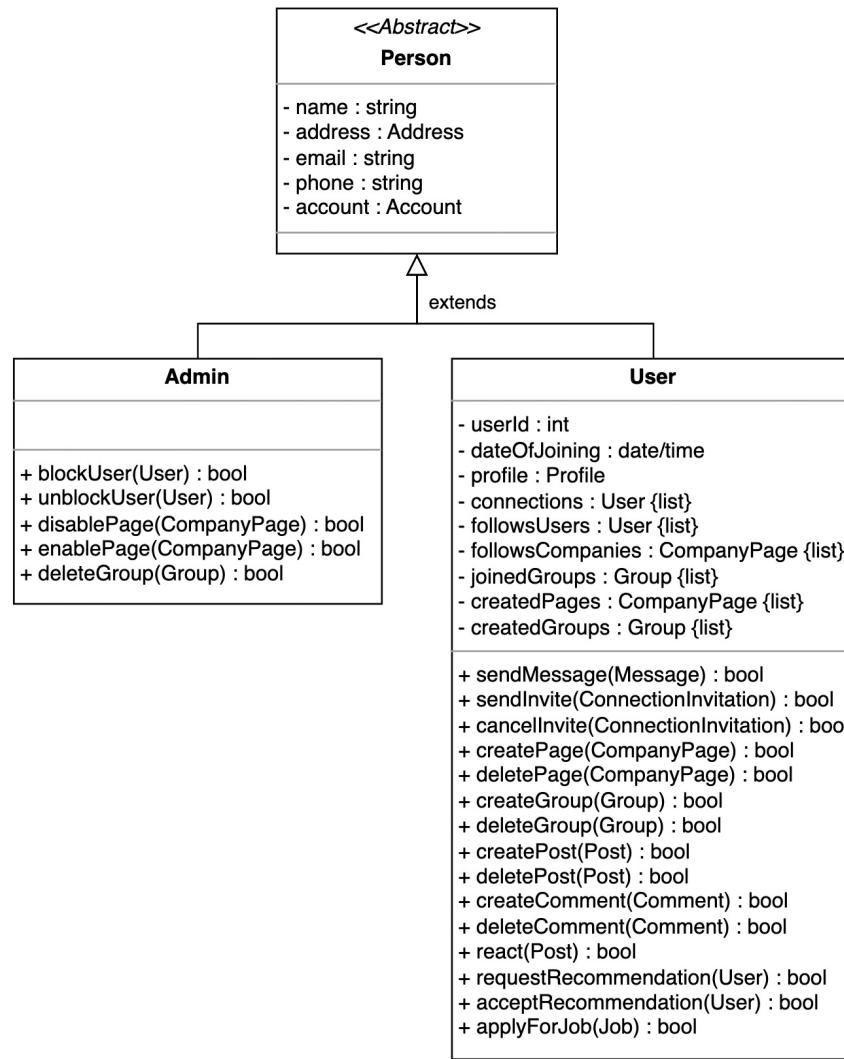
6. Post, Comment & Message

Post	Comment	Message
<ul style="list-style-type: none"> - postId : int - postOwner : User - text : string - media : byte {list} - totalReacts : int - totalShares : int - comments : Comment {list} <ul style="list-style-type: none"> + updateText() : bool 	<ul style="list-style-type: none"> - commentId : int - commentOwner : User - text : string - totalReacts : int - comments : Comment {list} <ul style="list-style-type: none"> + updateText() : bool 	<ul style="list-style-type: none"> - messageId : int - sender : User - recipients : User {list} - text : string - media : byte {list} <ul style="list-style-type: none"> + addRecipients(User {list}) : bool

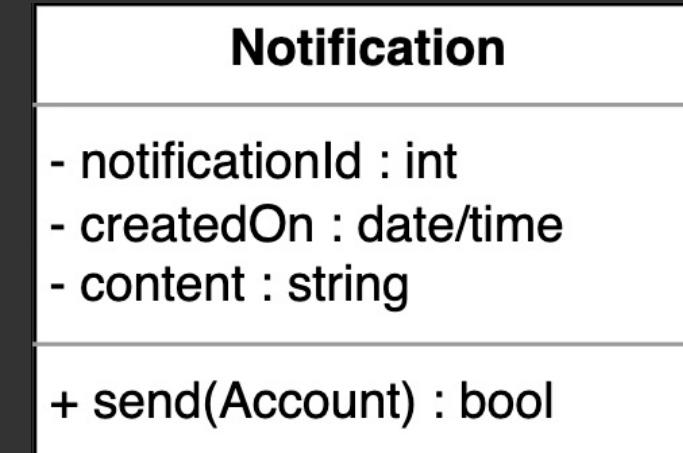
7. Connection Invitation

ConnectionInvitation
<ul style="list-style-type: none"> - sender : User - recipient : User - dateCreated : date/time - status : ConnectionInviteStatus <ul style="list-style-type: none"> + acceptConnection() : bool + ignoreConnection() : bool

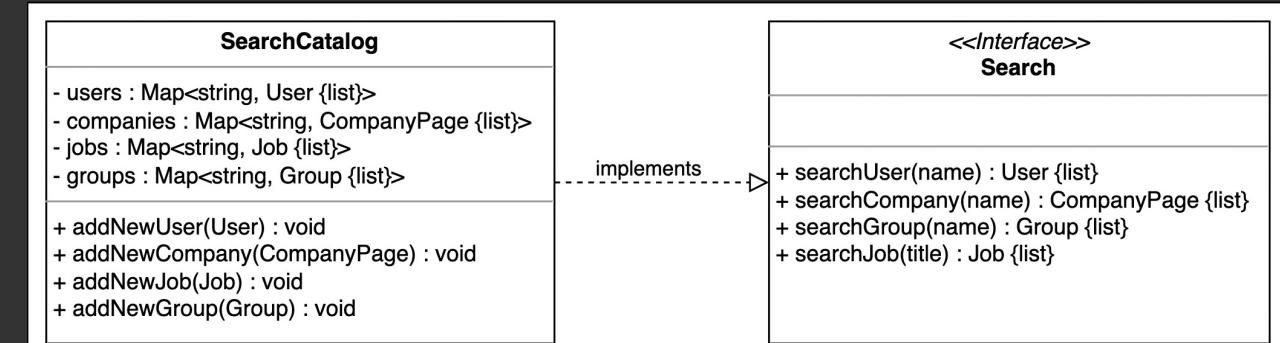
8 Person



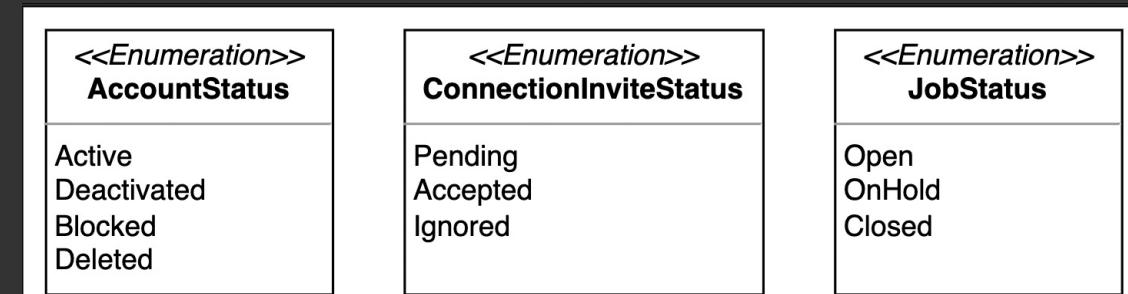
9. Notification



10. Search Interface & Search Catalog



11. Enumerations

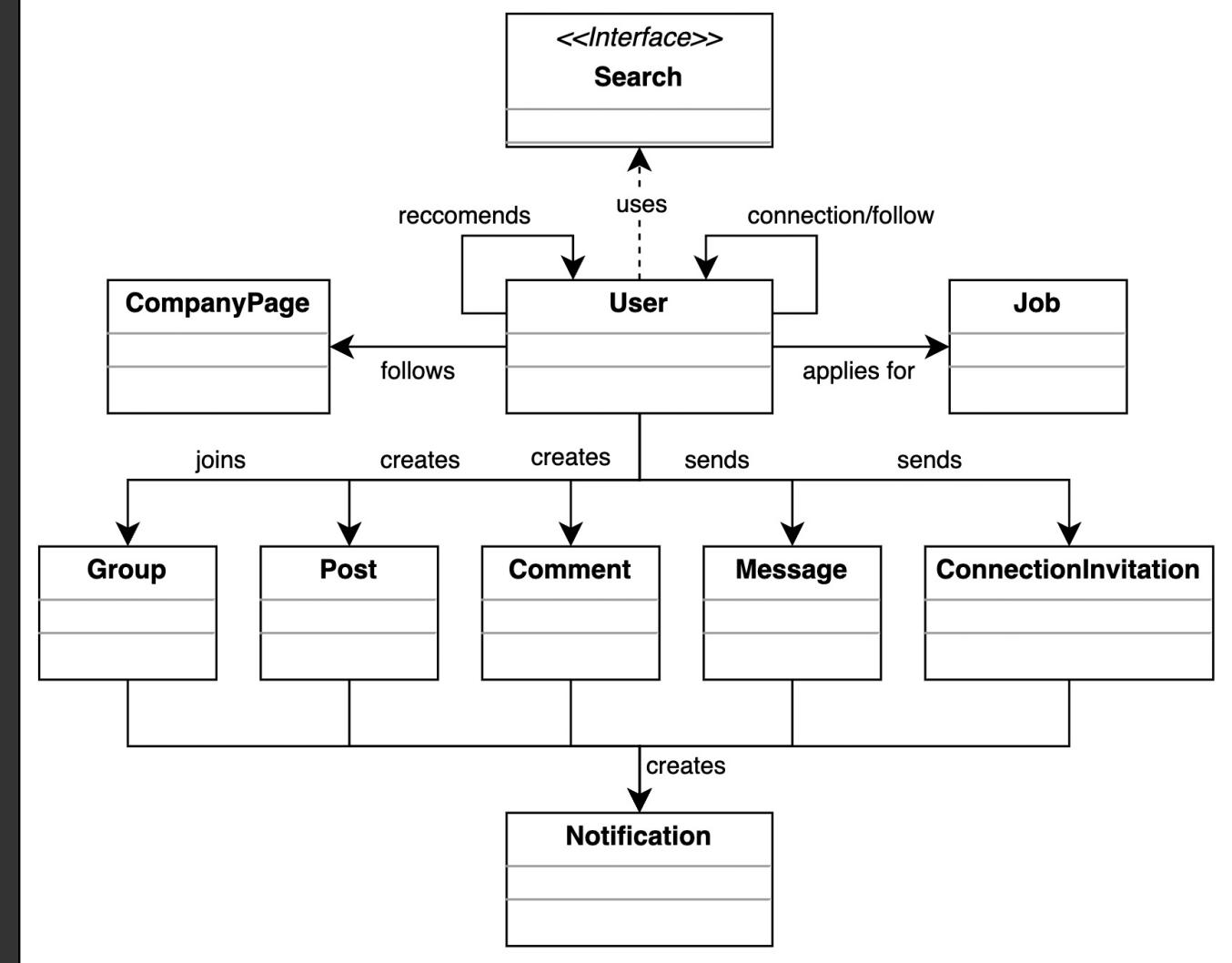


2. Address

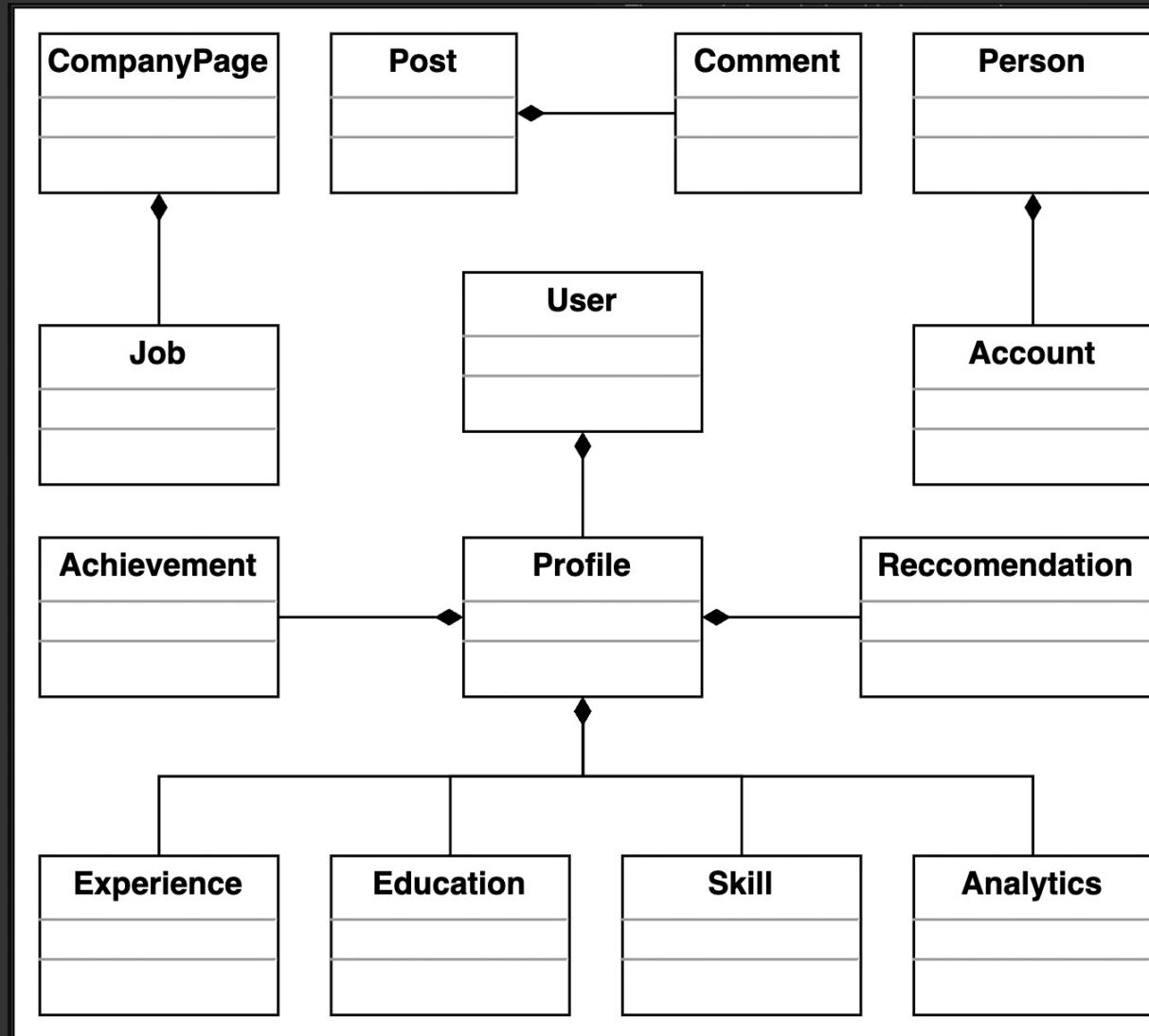
Address

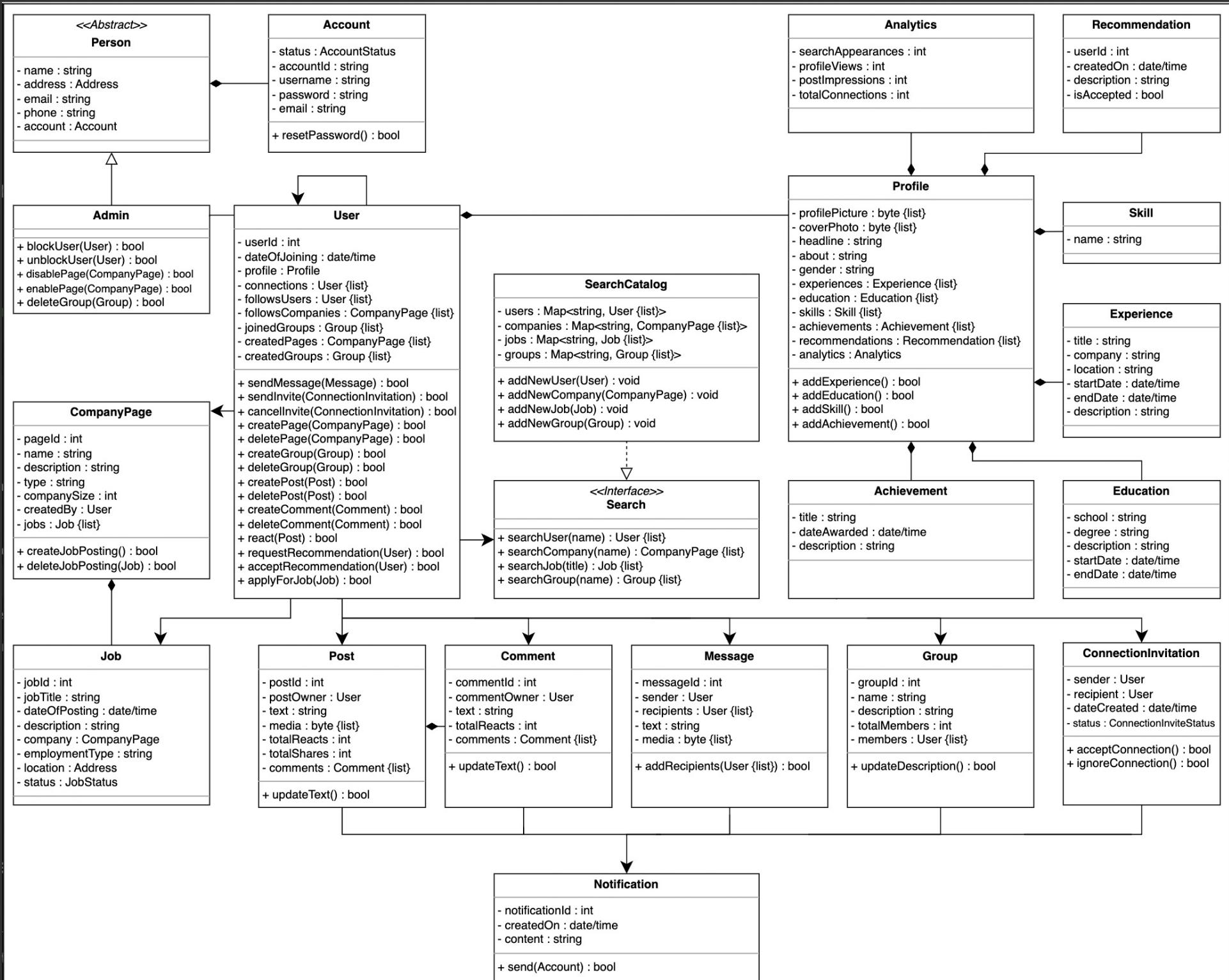
- zipCode : int
- streetAddress : string
- city : string
- state : string
- country : string

Association

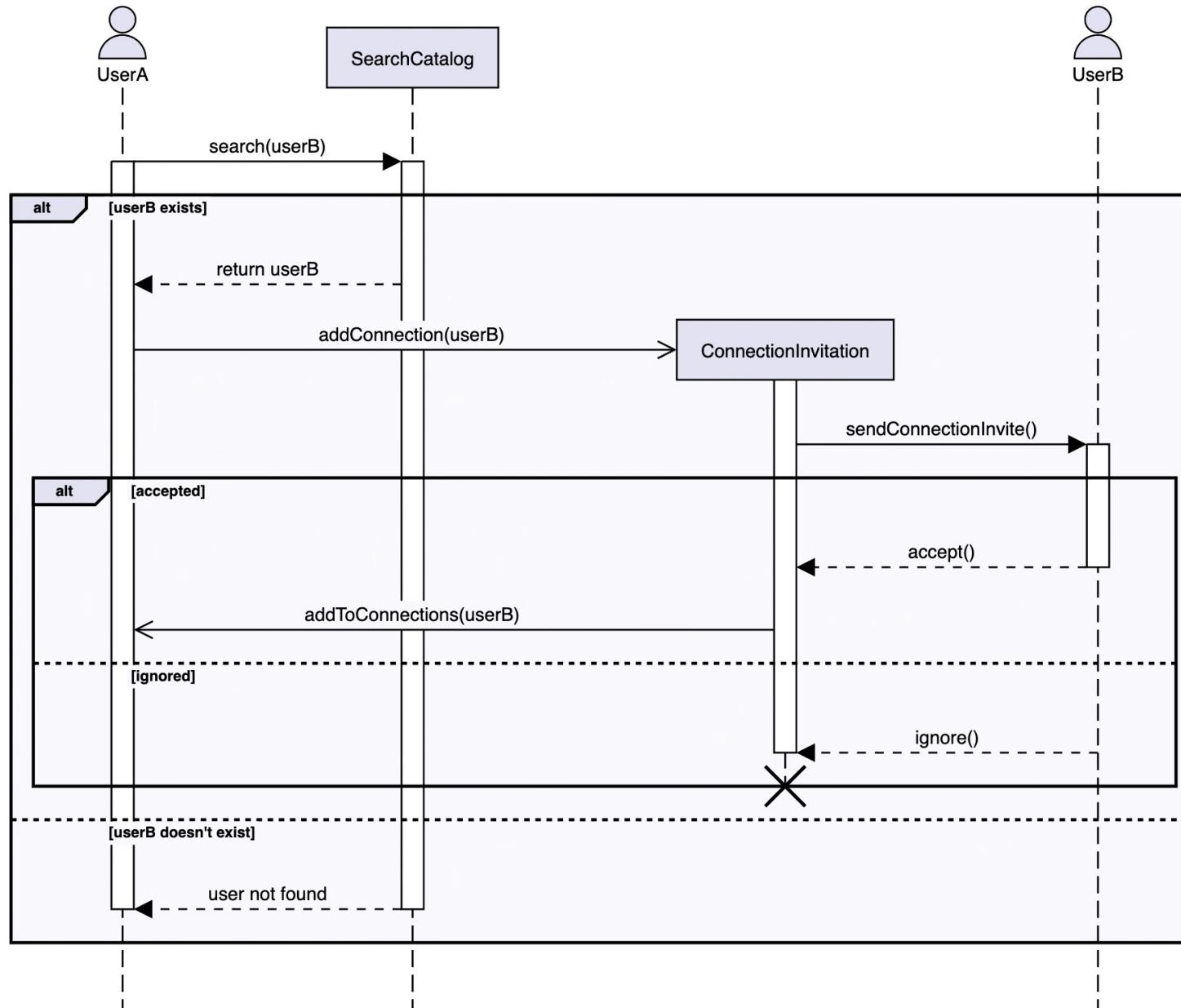


Composition



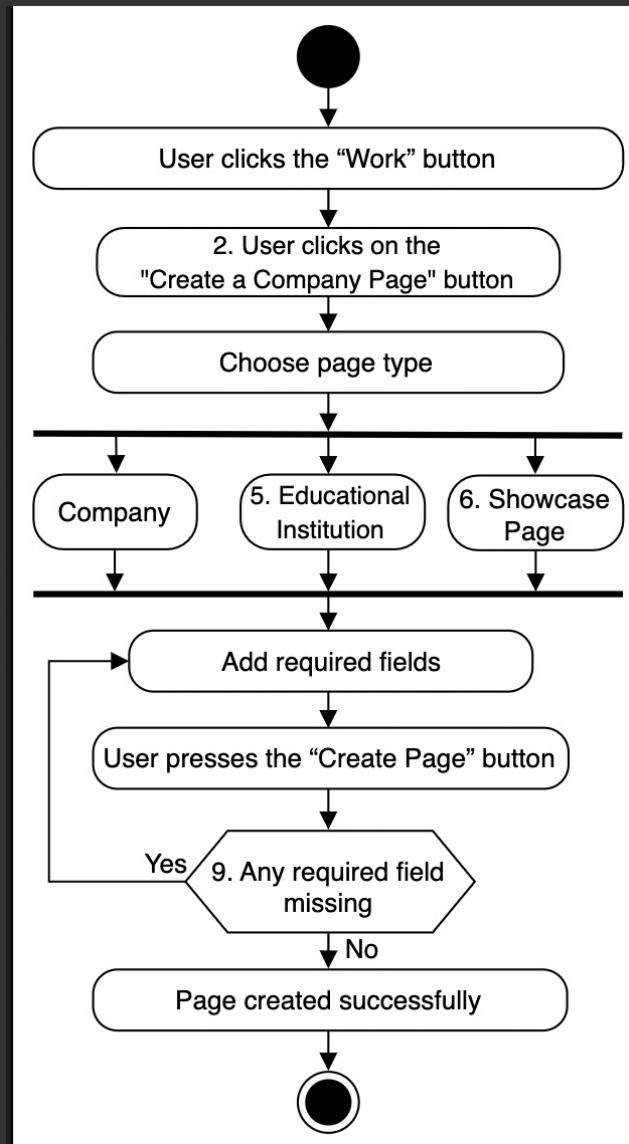
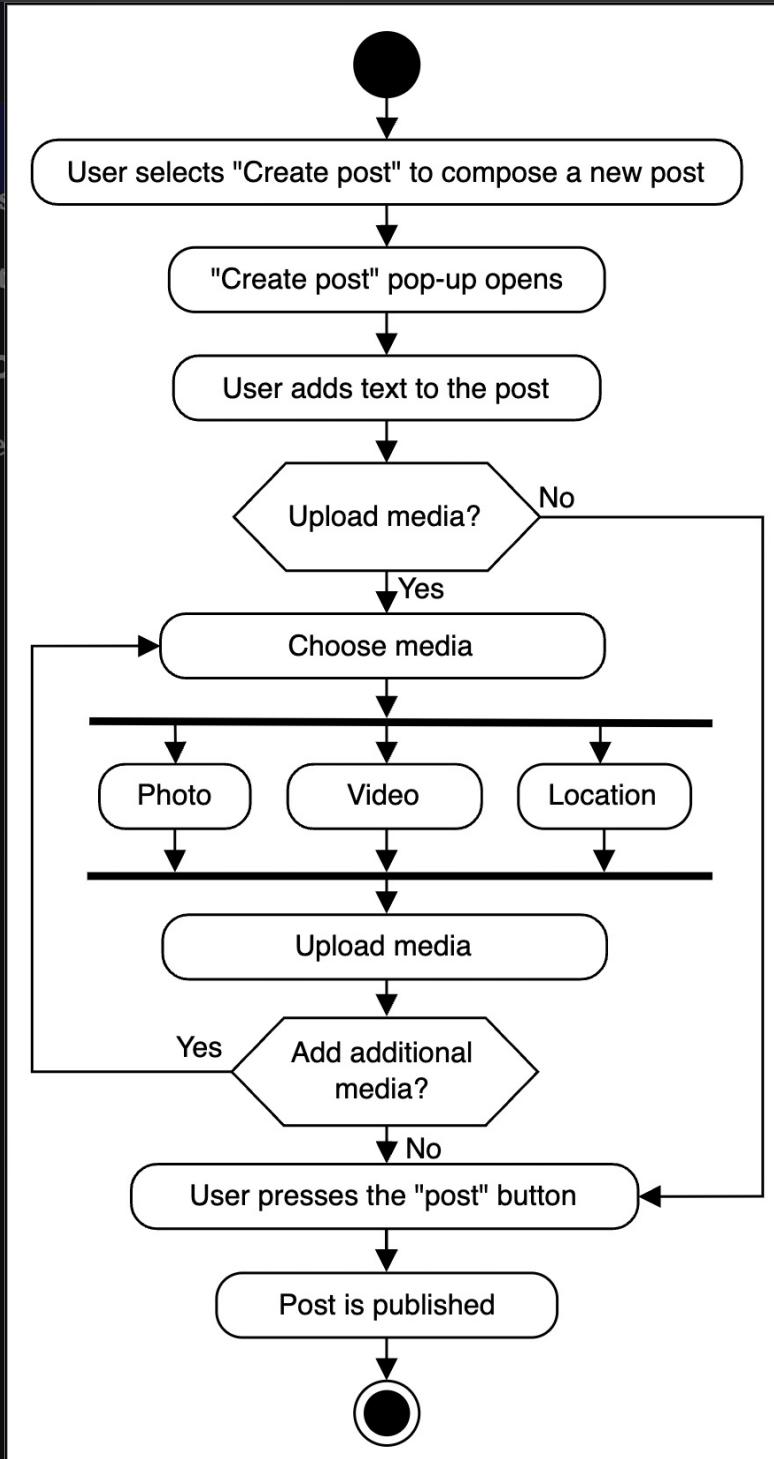


sd connection invitation



Connection
invitation

Actions



Creating a page

1. Constant

```
public class Address {  
    private int zipCode;  
    private String streetAddress;  
    private String city;  
    private String state;  
    private String country;  
}  
  
enum AccountStatus {  
    ACTIVE,  
    DEACTIVATED,  
    BLOCKED,  
    DELETED  
}  
  
enum ConnectionInviteStatus {  
    PENDING,  
    ACCEPTED,  
    IGNORED  
}  
  
enum JobStatus {  
    OPEN,  
    ON_HOLD,  
    CLOSED  
}
```

Code 2. Account

```
public class Account {  
    private String accountId;  
    private String password;  
    private String username;  
    private String email;  
    private AccountStatus status;  
  
    public boolean resetPassword();  
}
```

3. Person, Admin & User

```
// Person will be an abstract class  
public abstract class Person {  
    private String name;  
    private Address address;  
    private String email;  
    private String phone;  
    private Account account;  
}  
  
public class Admin extends Person {  
    public boolean blockUser(User user);  
    public boolean unblockUser(User user);  
    public boolean disablePage(CompanyPage page);  
    public boolean enablePage(CompanyPage page);  
    public boolean deleteGroup(Group group);  
}  
  
public class User extends Person {  
    private int userId;  
    private Date dateOfJoining;  
    private Profile profile;  
    private List<User> connections;  
    private List<User> followsUsers;  
    private List<CompanyPage> followCompanies;  
    private List<Group> joinedGroups;  
    private List<CompanyPage> createdPages;  
    private List<Group> createdGroups;  
  
    public boolean sendMessage(Message message);  
    public boolean sendInvite(ConnectionInvitation invite);  
    public boolean cancelInvite(ConnectionInvitation invite);  
    public boolean createPage(CompanyPage page);  
    public boolean deletePage(CompanyPage page);  
    public boolean createGroup(Group group);  
    public boolean deleteGroup(Group group);  
    public boolean createPost(Post post);  
    public boolean deletePost(Post post);  
    public boolean createComment(Comment comment);  
    public boolean deleteComment(Comment comment);  
    public boolean react(Post post);  
    public boolean requestRecommendation(User user);  
    public boolean acceptRecommendation(User user);  
    public boolean applyForJob(Job job);  
}
```

4. Recommendation, achievement & analysis

```
public class Recommendation {  
    private int userId;  
    private Date createdOn;  
    private String description;  
    private boolean isAccepted;  
}  
  
public class Achievement {  
    private String title;  
    private Date dateAwarded;  
    private String description;  
}  
  
public class Analytics {  
    private int searchAppearances;  
    private int profileViews;  
    private int postImpressions;  
    private int totalConnections;  
}
```

5. Profil, experience, Education & Skill

```
public class Experience {  
    private String title;  
    private String company;  
    private Address location;  
    private Date startDate;  
    private Date endDate;  
    private String description;  
}  
  
public class Education {  
    private String school;  
    private String degree;  
    private Date startDate;  
    private Date endDate;  
    private String description;  
}  
  
public class Skill {  
    private String name;  
}  
  
public class Profile {  
    private String headline;  
    private String about;  
    private String gender;  
    private List<byte> profilePicture;  
    private List<byte> coverPhoto;  
  
    private List<Experience> experiences;  
    private List<Education> educations;  
    private List<Skill> skills;  
    private List<Achievement> achievements;  
    private List<Recommendation> recommendations;  
    private Analytics analytics;  
  
    public boolean addExperience(Experience experience);  
    public boolean addEducation(Education education);  
    public boolean addSkill(Skill skill);  
    public boolean addAchievement(Achievement achievement);  
}
```

b. Company, Job & group

```
public class Job {  
    private int jobId;  
    private String jobTitle;  
    private Date dateOfPosting;  
    private String description;  
    private CompanyPage company;  
    private String employmentType;  
    private Address location;  
    private JobStatus status;  
}  
  
public class CompanyPage {  
    private int pageId;  
    private String name;  
    private String description;  
    private String type;  
    private int companySize;  
    private User createdBy;  
    private List<Job> jobs;  
  
    public boolean createJobPosting();  
    public boolean deleteJobPosting(Job job);  
}  
  
public class Group {  
    private int groupId;  
    private String name;  
    private String description;  
    private int totalMembers;  
    private List<User> members;  
  
    public boolean updateDescription();  
}
```

2 Post, comment, message & connection invitation

```
public class Post {  
    private int postId;  
    private User postOwner;  
    private String text;  
    private List<byte> media;  
    private int totalReacts;  
    private int totalShares;  
    private List<Comment> comments;  
  
    public boolean updateText();  
}  
  
public class Comment {  
    private int commentId;  
    private User commentOwner;  
    private String text;  
    private int totalReacts;  
    private List<Comment> comments;  
  
    public boolean updateText();  
}  
  
public class Message {  
    private int messageId;  
    private User sender;  
    private List<User> recipients;  
    private String text;  
    private List<byte> media;  
  
    public boolean addRecipients(List<User> recipients);  
}  
  
public class ConnectionInvitation {  
    private User sender;  
    private User recipients;  
    private Date dateCreated;  
    private ConnectionInviteStatus status;  
  
    public boolean acceptConnection();  
    public boolean rejectConnection();  
}
```

8. Search, Catalog & Notification

```
public interface Search {  
    // Interface method (does not have a body)  
    public List<User> searchUser(String name);  
    public List<CompanyPage> searchCompany(String name);  
    public List<Group> searchGroup(String name);  
    public List<Job> searchJob(String title);  
}  
  
public class SearchCatalog implements Search {  
    private HashMap<String, List<User>> users;  
    private HashMap<String, List<CompanyPage>> companies;  
    private HashMap<String, List<Group>> groups;  
    private HashMap<String, List<Job>> jobs;  
  
    public void addNewUser(User user);  
    public void addNewCompany(CompanyPage company);  
    public void addNewGroup(Group group);  
    public void addNewJob(Job job);  
  
    public List<User> searchUser(String name) {  
        // functionality  
    }  
  
    public List<CompanyPage> searchCompany(String name) {  
        // functionality  
    }  
  
    public List<Job> searchJobs(String title) {  
        // functionality  
    }  
  
    public List<Group> searchGroups(String name) {  
        // functionality  
    }  
}  
  
public class Notification {  
    private int notificationId;  
    private Date createdOn;  
    private String content;  
  
    public boolean sendNotification(Account account);  
}
```