Spotagram
Design Review

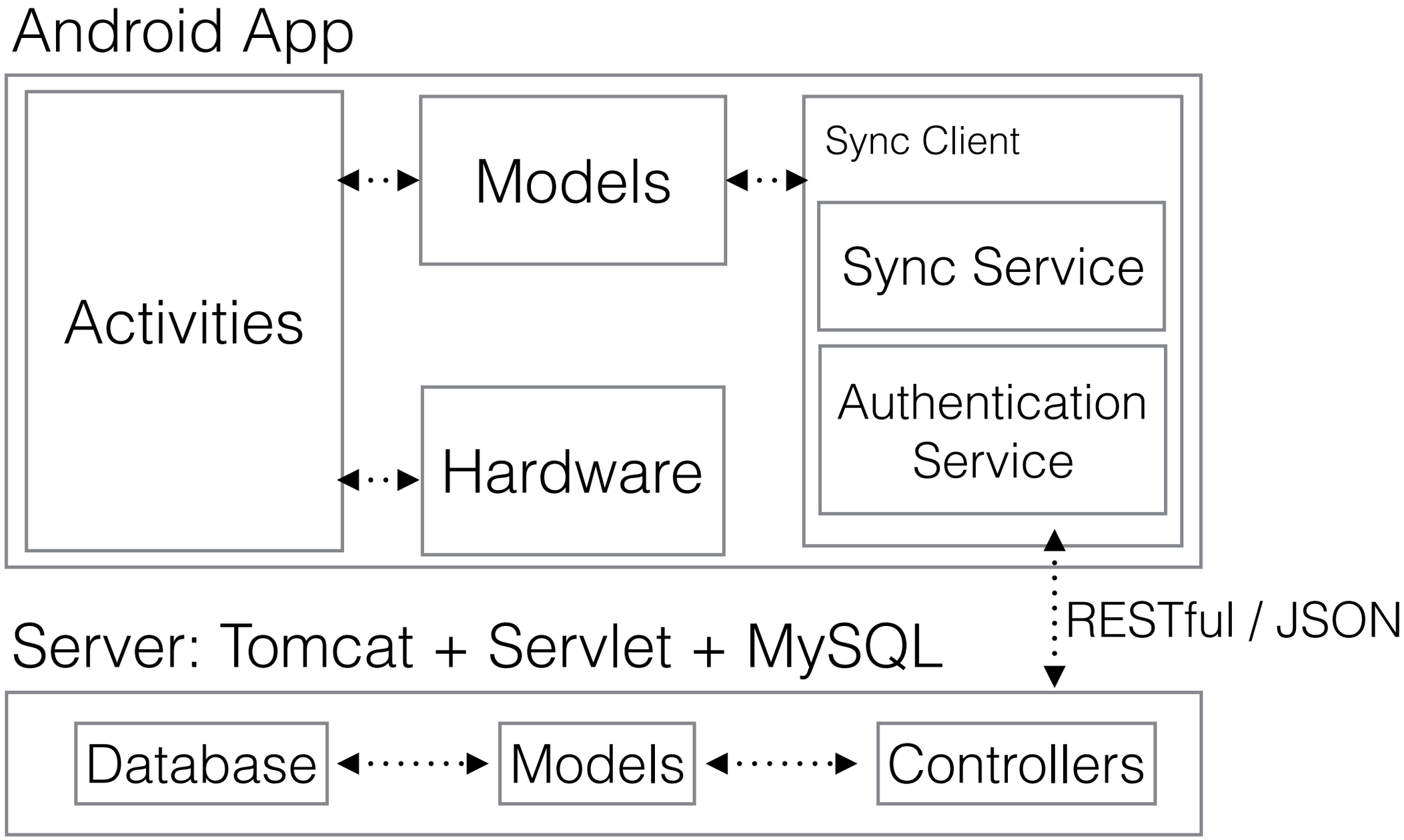
leiyu miaojunl dawang
20 Nov 2015

User Story

- Share your **POSITION** with everyone (with words/images)
- See what's **HAPPENING** nearby on the map
- Use **TAG** to see what you want (Lost&Found, Accident, Restaurant Reviews, etc)

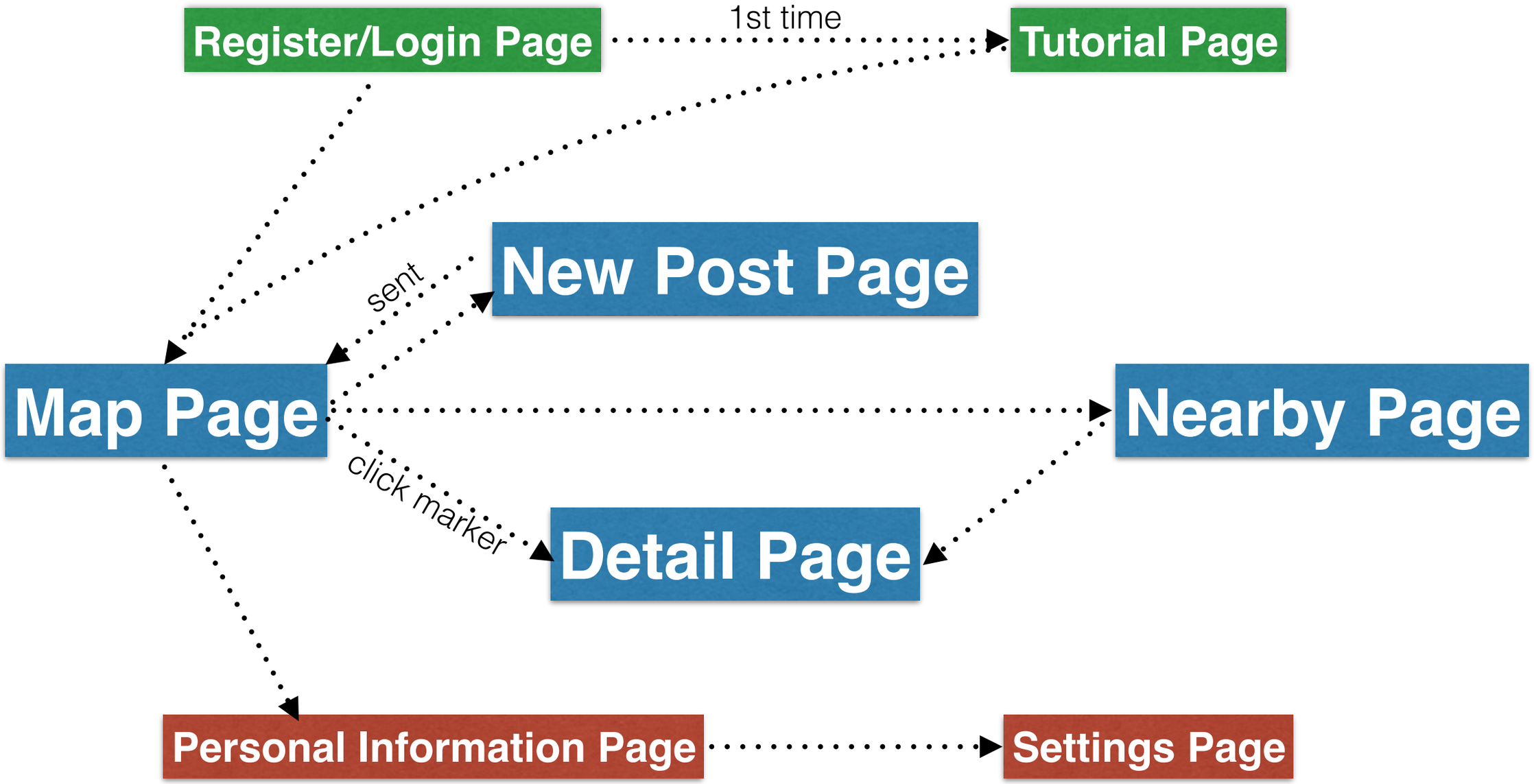
General

- **Location** Based Social App
- Be yourself! **No** follow/unfollow relationship. You can be friends when visiting the same place
- Similar to Instagram, instead of sharing photos, we **share locations**
- User: Students, Drivers, Teacher -> basically **everyone**!



Architecture

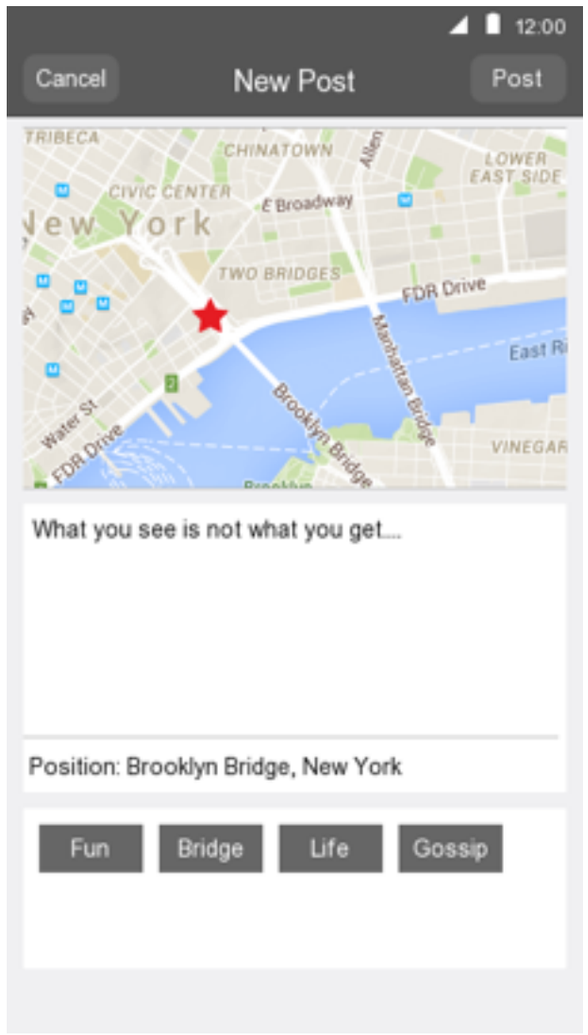
Overview of Client/Server Design
Communicate with RESTful API/JSON



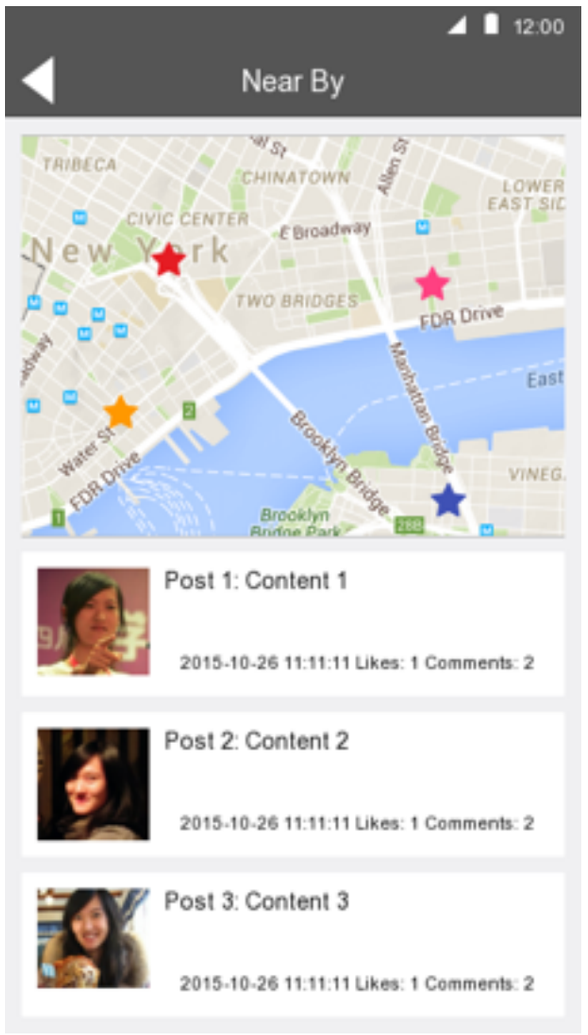
Page Flow

Layers indicate with color
Not all relations for pages are shown

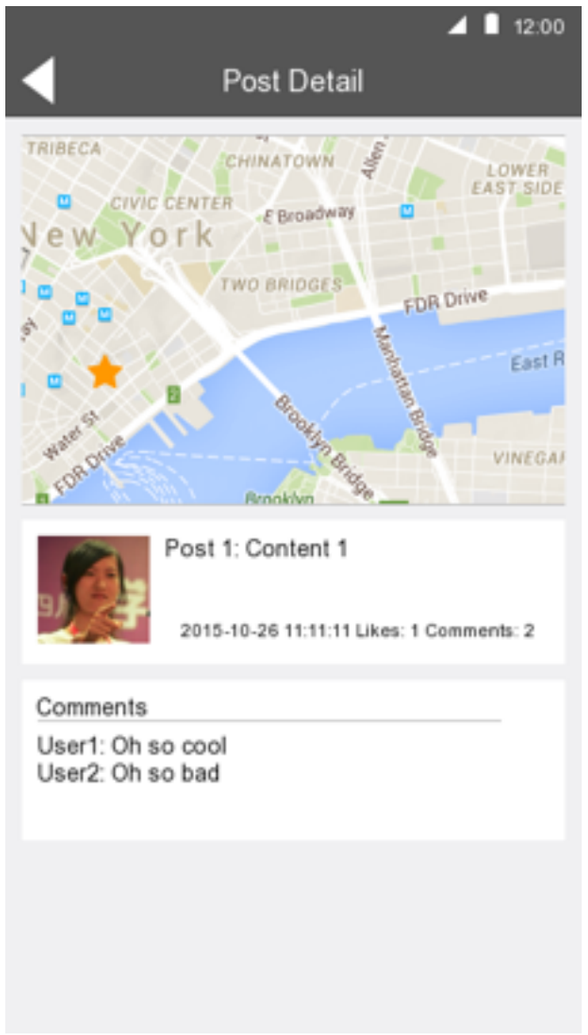
Main Page



Post New Note



Nearby



Note Detail

Presentation Tier

Not all pages are shown here
Introduce the main features

Note

- id, user, time, date, location, content, comment
- use Collections to store the data

User

- id, username, password, gender, email

Comment

- id, user, time, note, content

Application Tier

Models: getter/setter and other basic validation method
Activities: methods for starting new intent or responding button clicking

Table

User Note Comment

Servlet

GetComment GetNotes

Login PostNote

Registration ReplyNote

Util

Save Load JSONGenerator

Server Tier

Layers indicate with color

Not all relations for pages are shown

Protocol of Registration

Name	Info
Description	In this protocol, the client send username, email and password to server to register.
Type	POST
Notice	The client should check the correctness of request type

Format of Request

Key	Type	Value
user	String	The username
email	String	User email
pwd	string	The password of user

Format of Response

Key	Type	Value
result	int	The result code about the request

Format of Result code

Key	Value	Description
RESULT_OK	0	Log in success
RESULT_USER_ERR	-1	User already existed
RESULT_EMAIL_ERR	-2	Email already existed

Protocol of getting notes

Name	Info
Description	In this protocol, the client send request to get the notes in specific area.
Type	POST
Notice	

Format of Request

Key	Type	Value
loc_longitude	float	Longitude of the location
loc_latitude	float	Latitude of the location
radius_km	int	The range of the area to get the notes
max_note	int	The maximum number of notes to get
start_note	int	The start index of notes to get (usually is 0)

Format of Response

Key	Type	Value
result	int	The result code about the request
note_lists	JSON	The notes returned

Format of Result code

Key	Value	Description
RESULT_OK	0	Log in success
RESULT_ERR	-3	Undefined

Format of JSON note

Key	Type	Description
user_id	int	Id
user_name	string	name
loc_longitude	float	Longitude of the location
loc_latitude	float	Latitude of the location
info	String	The information of note
date	date	undefined

Protocol of getting comments from a note

Name	Info
Description	In this protocol, the client get comments related to one note
Type	POST
Notice	**null**

Format of Request

Key	Type	Value
note_id	int	Note id

Format of Response

Key	Type	Value
result	int	The result code about the request

Format of Result code

Key	Value	Description
RESULT_OK	0	Log in success
RESULT_ERR	-3	Undefined

Format of JSON note

Key	Type	Description
comment_id	int	Comment id
user_id	string	User id
user_name	float	User name
content	String	The information of note
date	datetime	undefined

Protocol

For the communication between client & server
Not all are shown

Note Table

Name	Type	Comment
_id	INT	AUTO_INCREMENT PRIMARY_KEY
longitude	FLOAT	longitude
latitude	FLOAT	Latitude
date	DATETIME	date of the post
content	TEXT	content of the post
type	INT	different type of post
userid	INT	FOREIGN_KEY
username	TEXT	name of the userid
info	TEXT	json description content

User Table

Name	Type	Comment
_id	INT	AUTO_INCREMENT PRIMARY_KEY
username	VARCHAR(30)	no longer than 30 chars
password	VARCHAR(30)	no longer than 30 chars
gender	BIT	male / female
avatar	TEXT	url address
info	TEXT	json description content

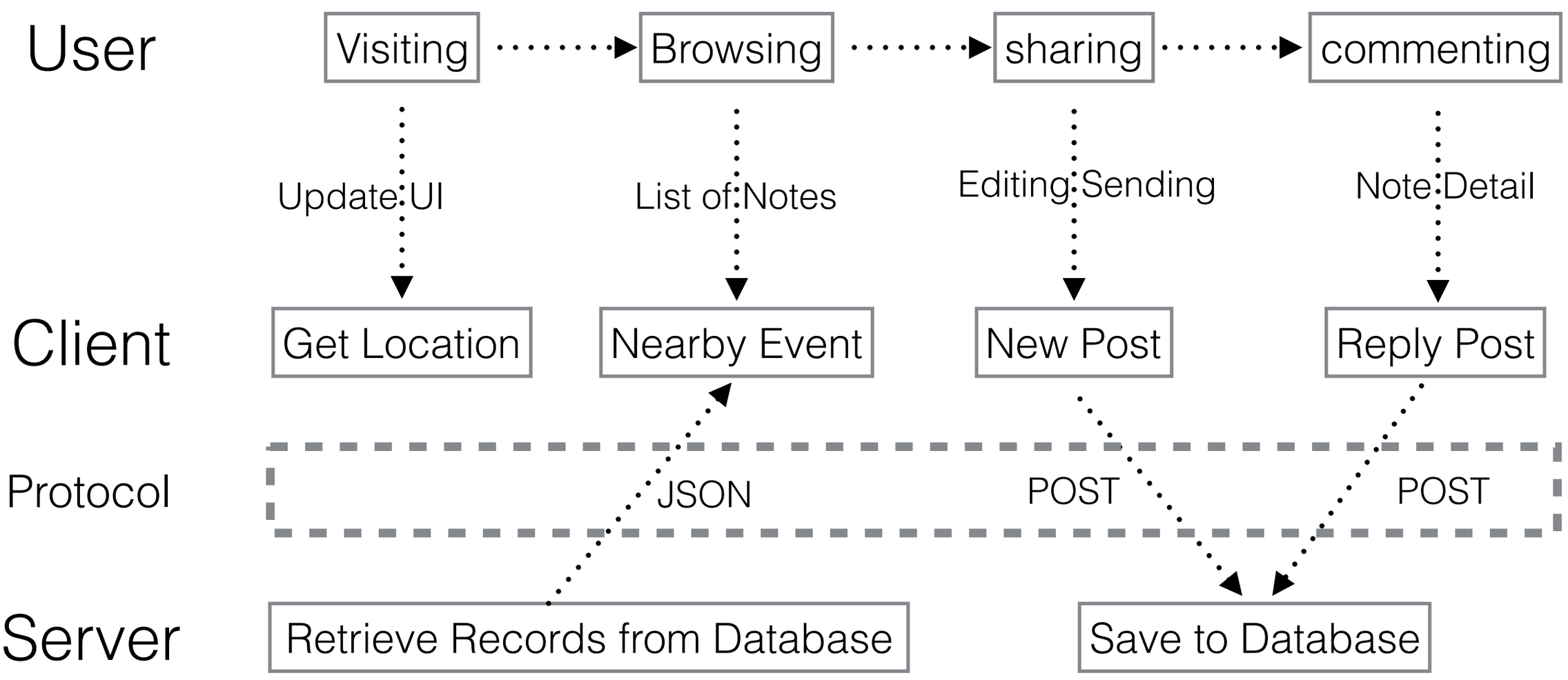
Comment Table

Name	Type	Comment
_id	INT	AUTO_INCREMENT PRIMARY_KEY
date	DATETIME	date of the post
content	TEXT	content of the post
userid	INT	FOREIGN_KEY (this is the user who posts the comment)
username	TEXT	the name of userid
noteid	INT	FOREIGN_KEY

Database

No Database on the client side

User Scenario



Thanks