

REPORT 2

PATH



PATH

Yu Yanjia

Zhao Yun

Zhao Binqi

Software Engineering, 2018 Spring Semester

Contents

Part 1

General Description

Prior Work -----2

To-Be Done ----- 3

Part 2

Object Oriented Design

Use Case Diagram ----- 4

System Sequence Diagram ----- 7

Class Diagram -----9

State Diagram -----10

Part 3

Logical Design

System Context Data Flow Diagram ---12&14

Functional Decomposition Diagram -----13

Event Diagram ----- 15

System Diagram -----17

Part One

General Description

Our project, PATH, has already made some progress since the last report, and now is undergoing the process of coding and testing on real cell phones. Applying the technique of Agile Process and Prototyping, our app is now updated about twice per week.

However, there still remain many problems to be solved and many errors to be debugged, furthermore, we are now currently involving ourselves in the learning of location-tracking and database, which are, by far, our biggest challenges.

Prior Work

We have already constructed the main User-Interface using the story board in Xcode and allow our initial users first actually utilize with the UI design. We have also successfully implanted the Map Kit inside our app and is currently being tested and debugged.

To-Be Done

In the following weeks that are to come, we will be committed to these following:

1, Prototyping will be accelerated, since the main function are not yet to be developed.

2, Further beautify UI design, simplify the logic and optimize the methods for location-tracking and logical design.

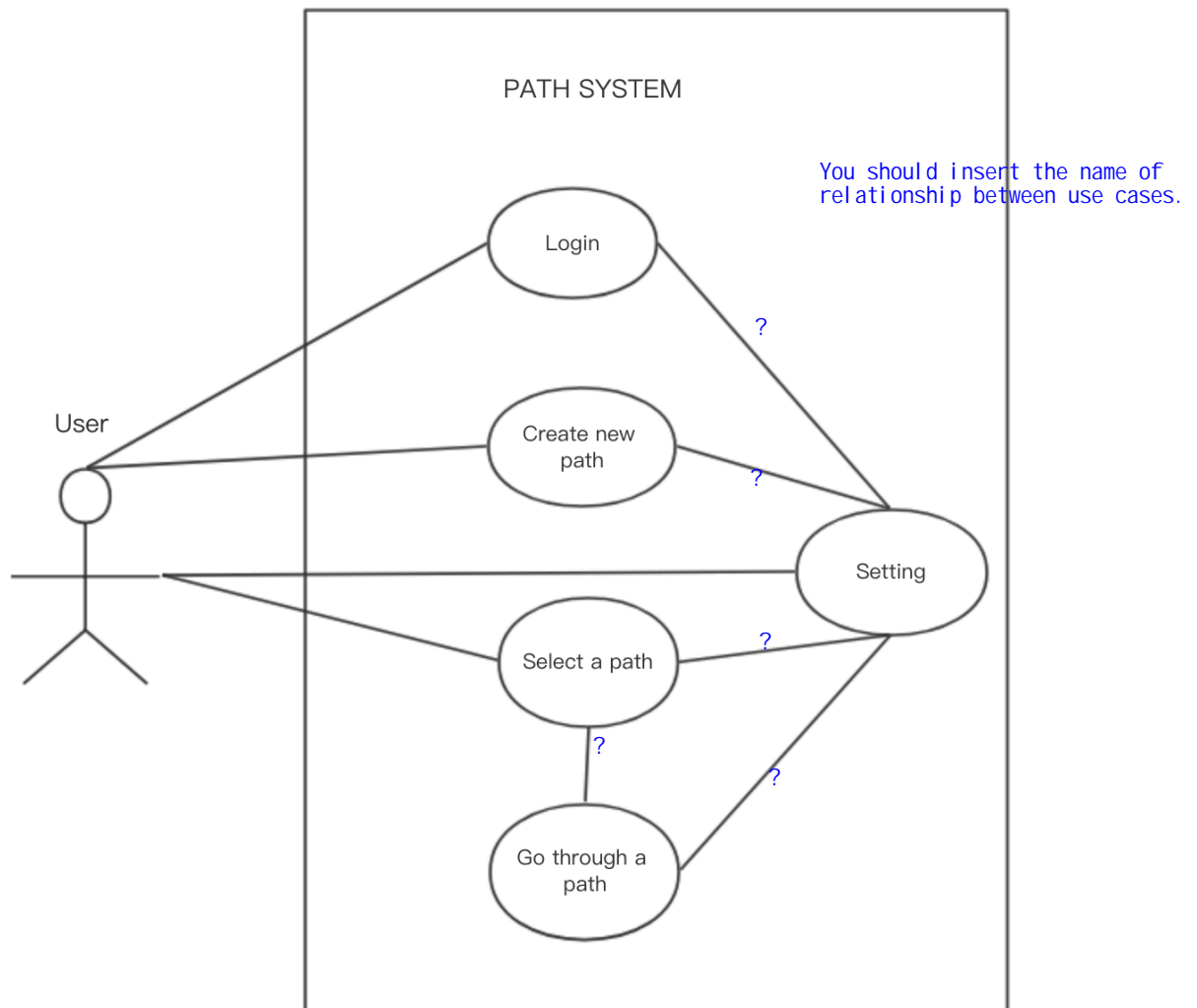
Part Two

Object-Oriented Design

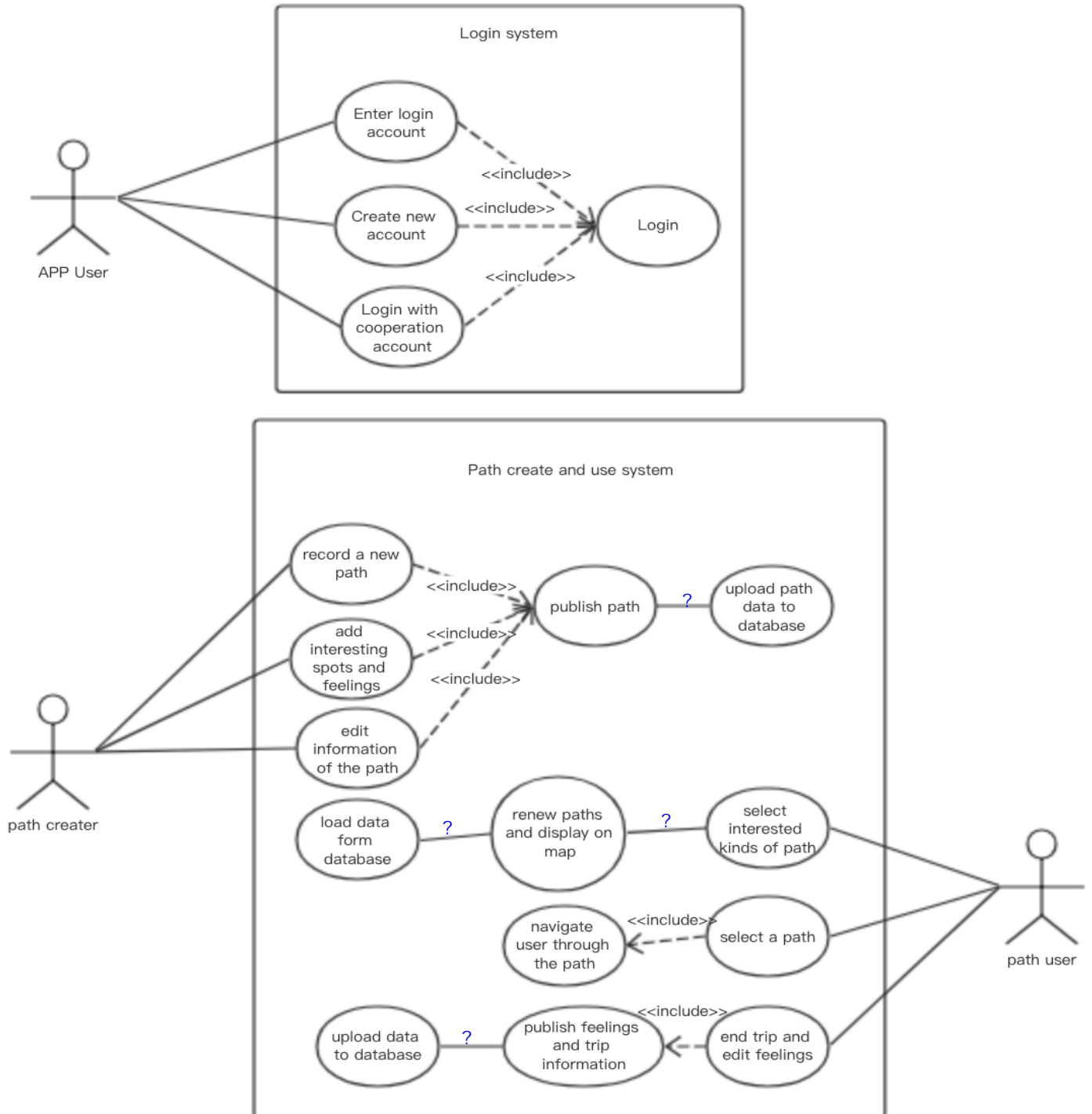
Use-Case Diagram (Abstract)

We believe that PATH can be divided into several use-cases, as report one indicated as below.

ACTOR	GOAL	USE CASE NAME
Creator & User	Log in app	Log in(UC-1)
Creator	Create a path	Create(UC-2)
Creator	Publish the path	Publish(UC-3)
User	Select one path and start trip	Select(UC-4)
User	End trip and see result and make comment	End trip(UC-5)

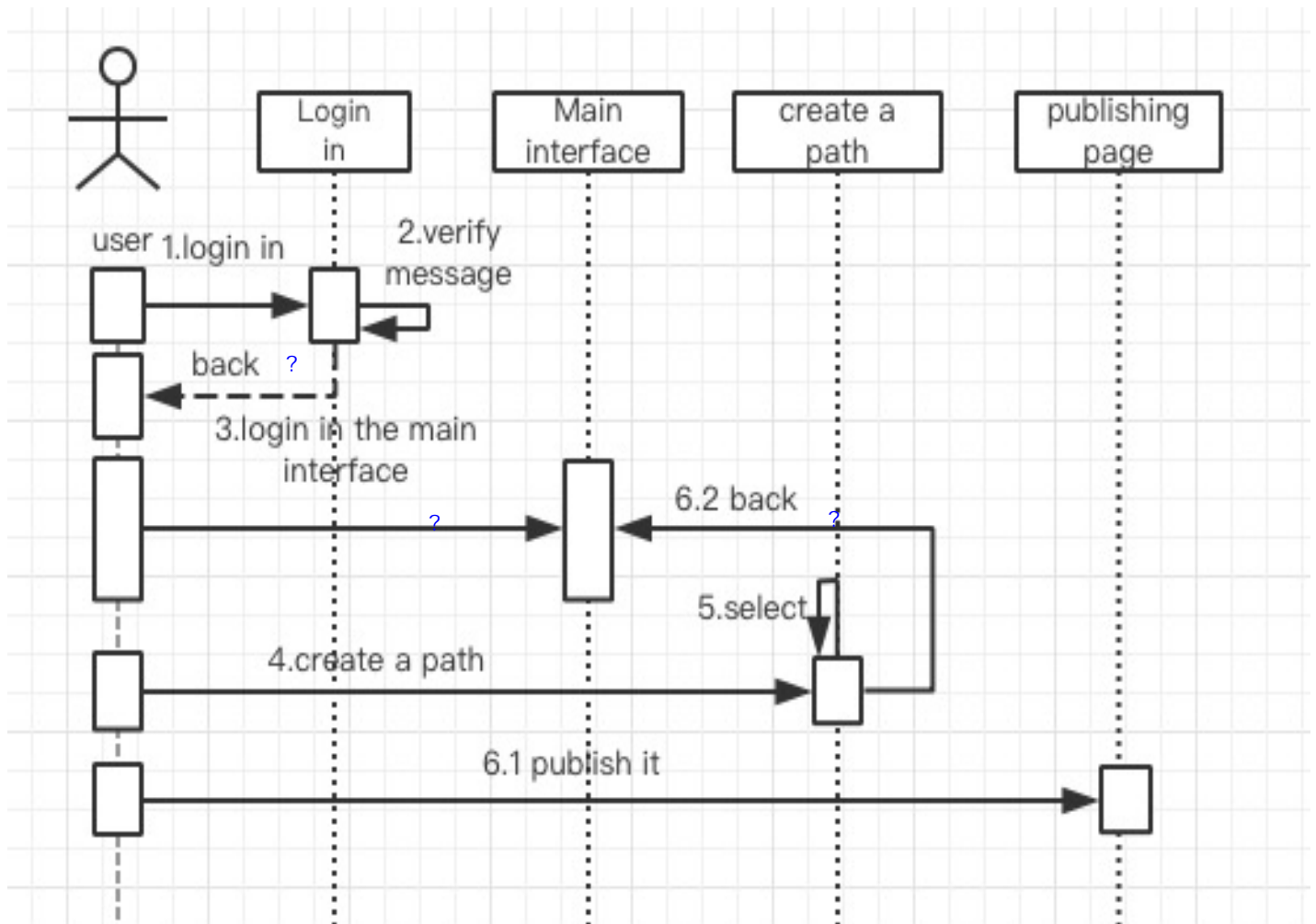


Use-Case Diagram (Full-Bodied)



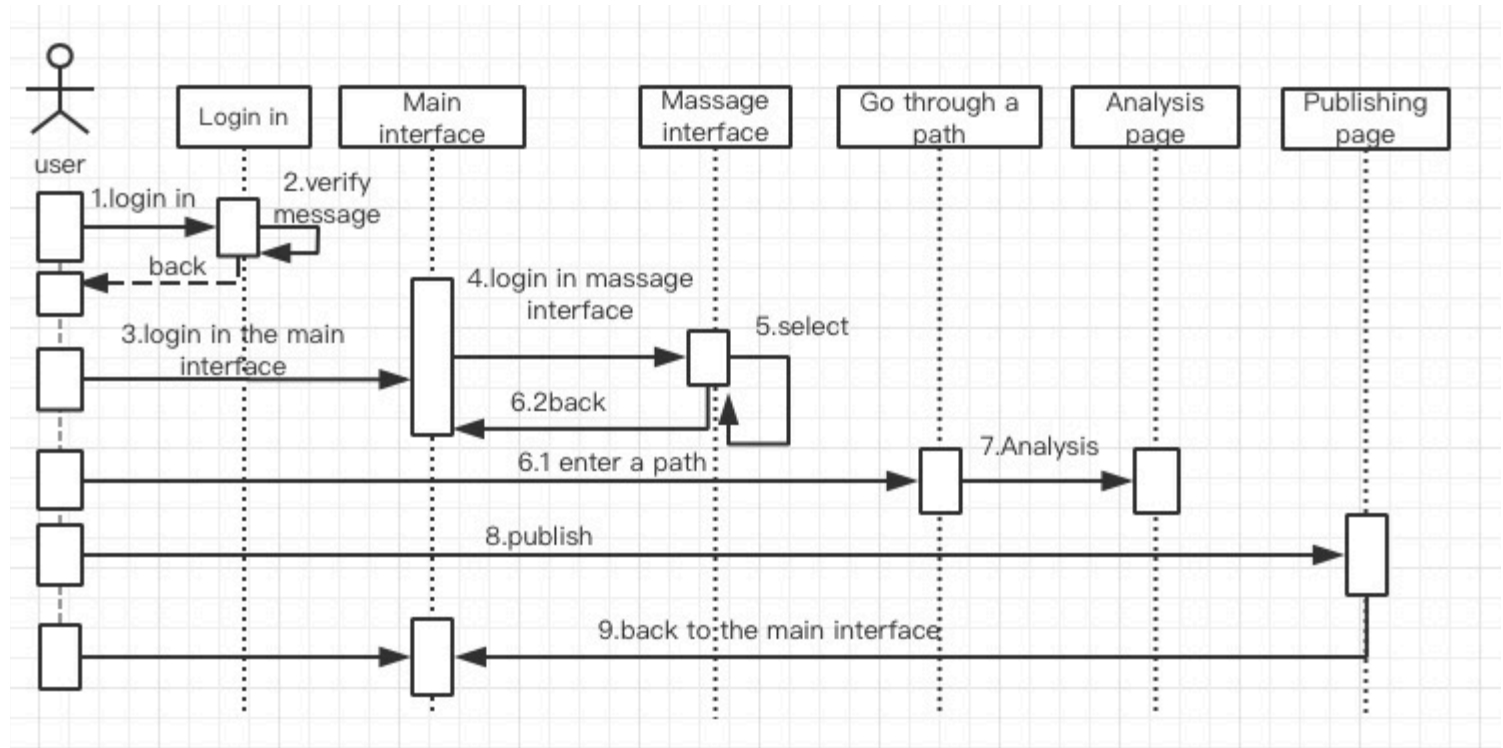
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

~~System~~ Sequence Diagram (Creating a Path)



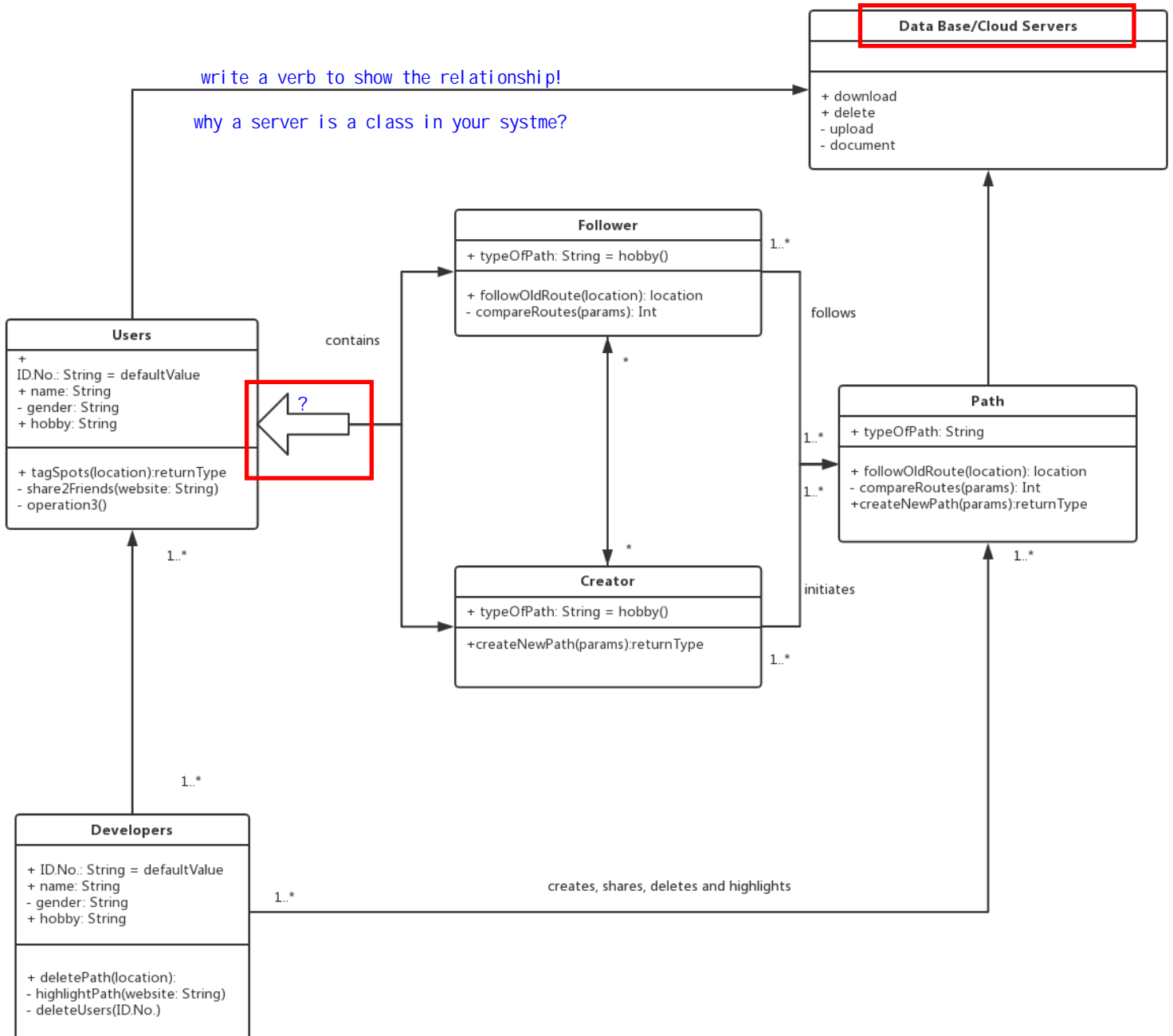
The sequence system describes the temporal order in which messages are sent between objects to show dynamic collaboration among multiple objects. It can represent the sequence of the use case behavior. When executing a use case behavior, each of these messages corresponds to a trigger event that causes a conversion in a class operation or state machine.

~~System~~ Sequence Diagram (Following a Path)



In our sequence diagram, we can see two different events that the user performed through the operation. One of them is: enter system-entry path-join path-analysis path, and the other is: enter system-create path-publish path.

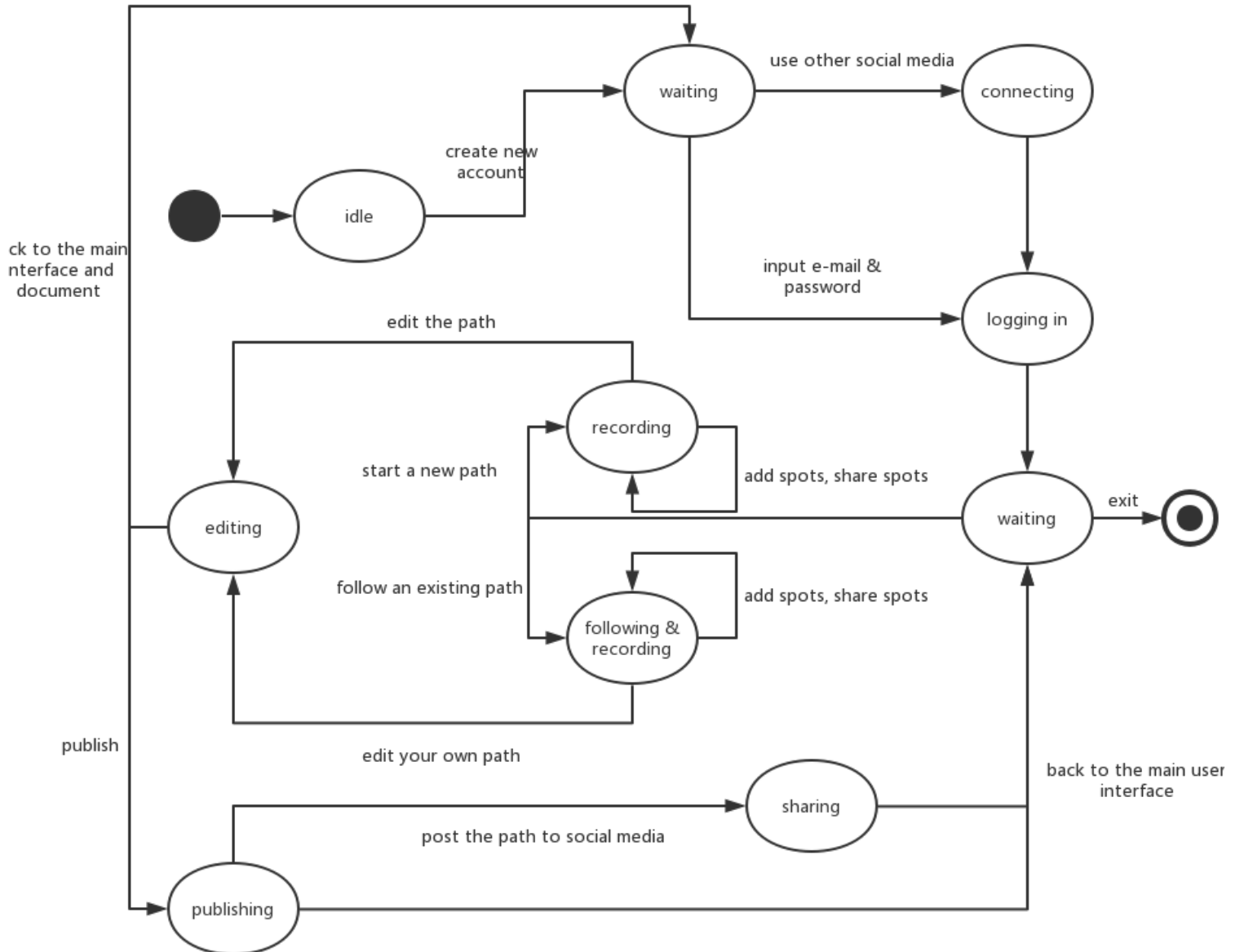
Class Diagram



We think that the main classes can be divided into these parts, users, developers, paths and outer data base/cloud servers. And the users are composed of two parts, which are the followers and the creators. This diagram clearly shows the relationship between the six components interacting with each other.

State Diagram

state diagram is used to show the states of some objects in the system, what's the object of this diagram.



When a user is first introduced to this app, the first thing to do is to either create a new account or log in to an existing account, after this step, the user is in the main user's interface, in which he can then choose to do the following, record a new route of his own or follow an existing route but at the same time, record his

own. During this process, the user, whether be the definer or the follower, can stop the recording at any given second and post and share his/her immediate emotion or tag a place that is very interesting or simply take a photo and then upload them to the route.

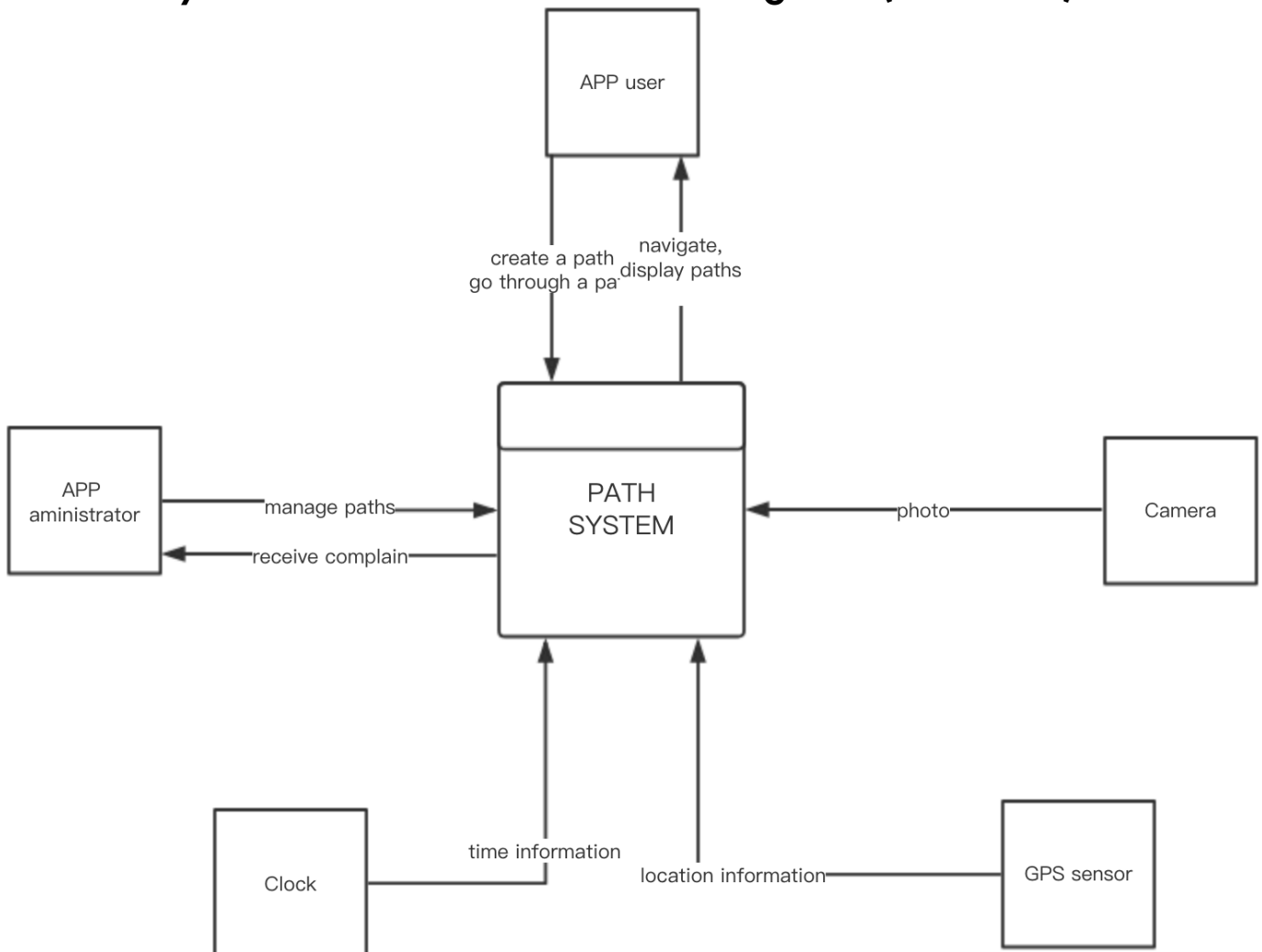
After finishing a route, one can easily edit the path and choose to either publish this path at the very instant or choose to save on the local drive.

When publishing, the user may also share the route to any other social media he/she likes, say, WeChat moments, Facebook or any other network.

Part Three

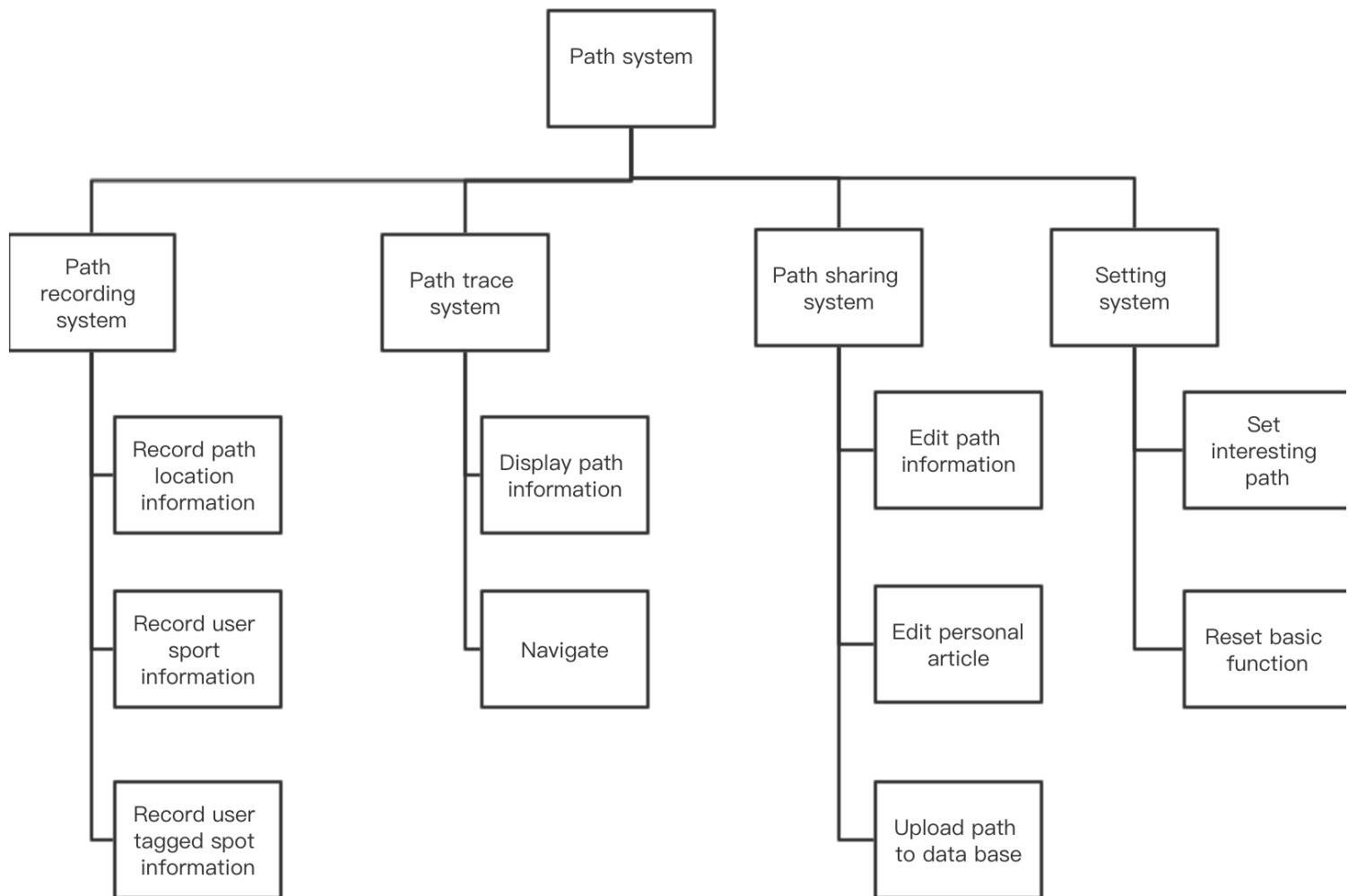
Logical Design

System Context Data Flow Diagram (Abstract)



A system context diagram in engineering is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high level view of a system. It is similar to a block diagram. In this figure, we can see the relationship between entities and interactions.

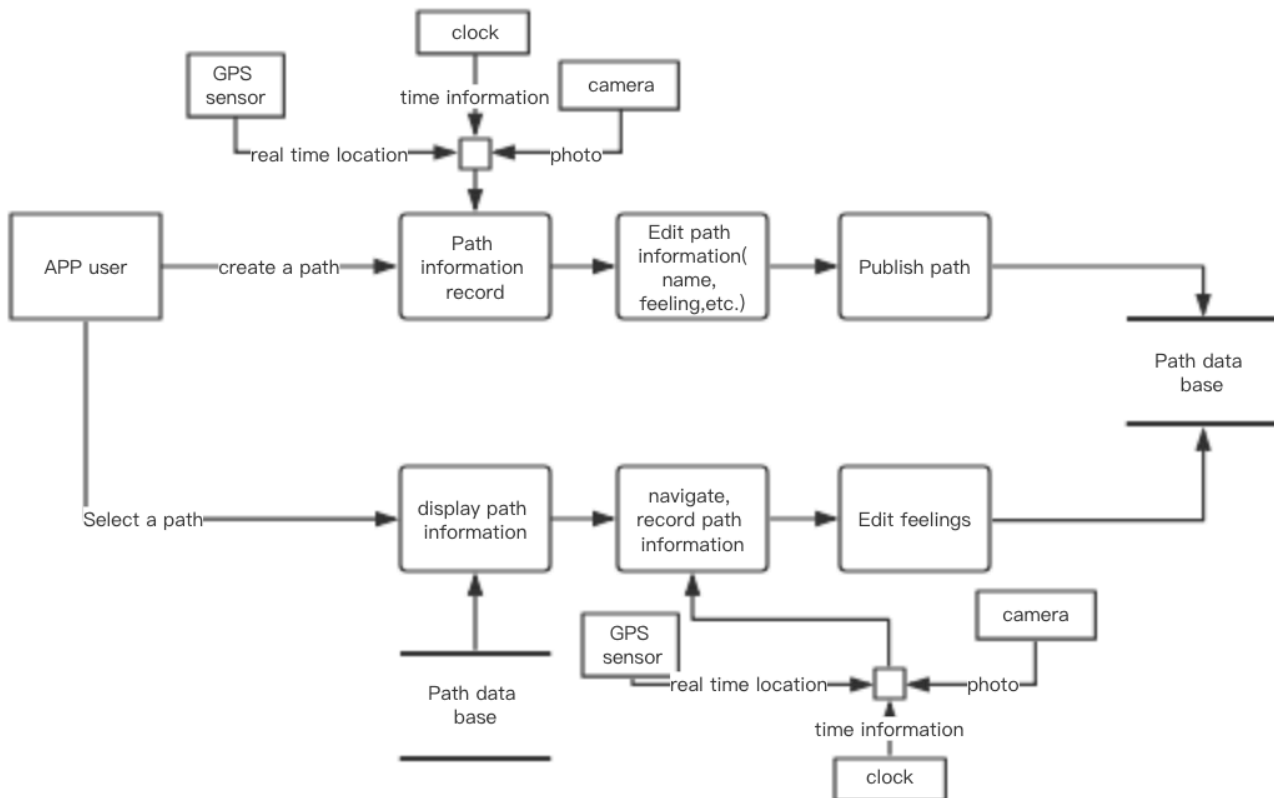
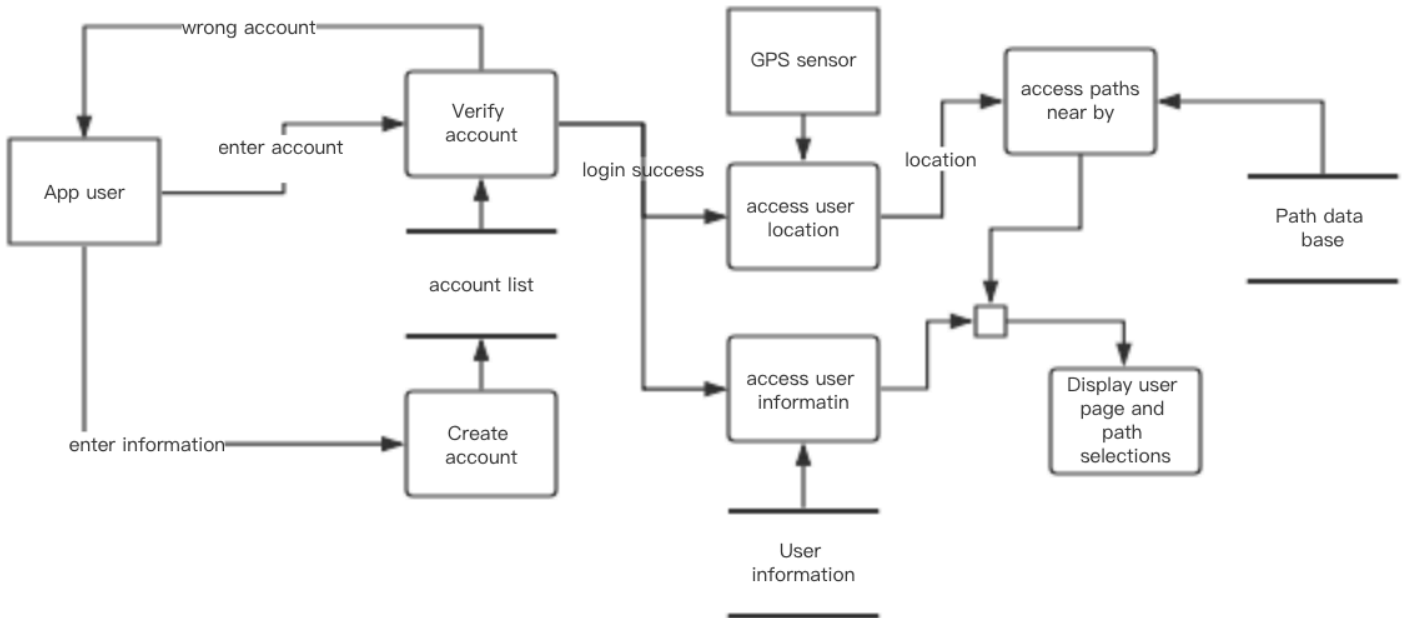
Functional Decomposition Diagram



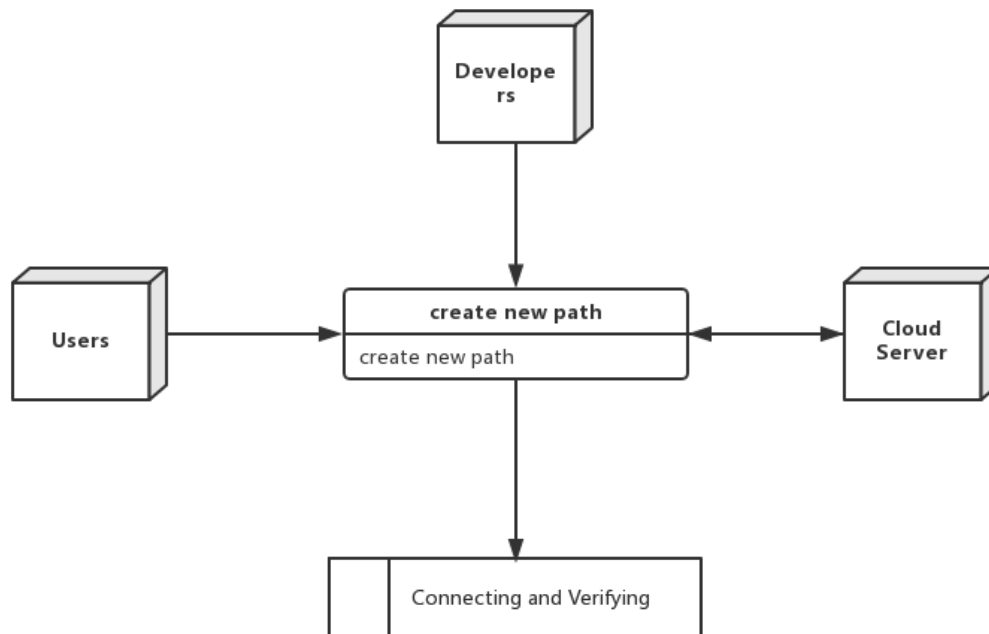
This process of decomposition may be undertaken to gain insight into the identity of the constituent components which may reflect individual physical processes of interest.

Interactions between the components are critical to the function of the collection. All interactions may not be observable, but possibly deduced through repetitive perception, synthesis, validation and verification of composite behavior.

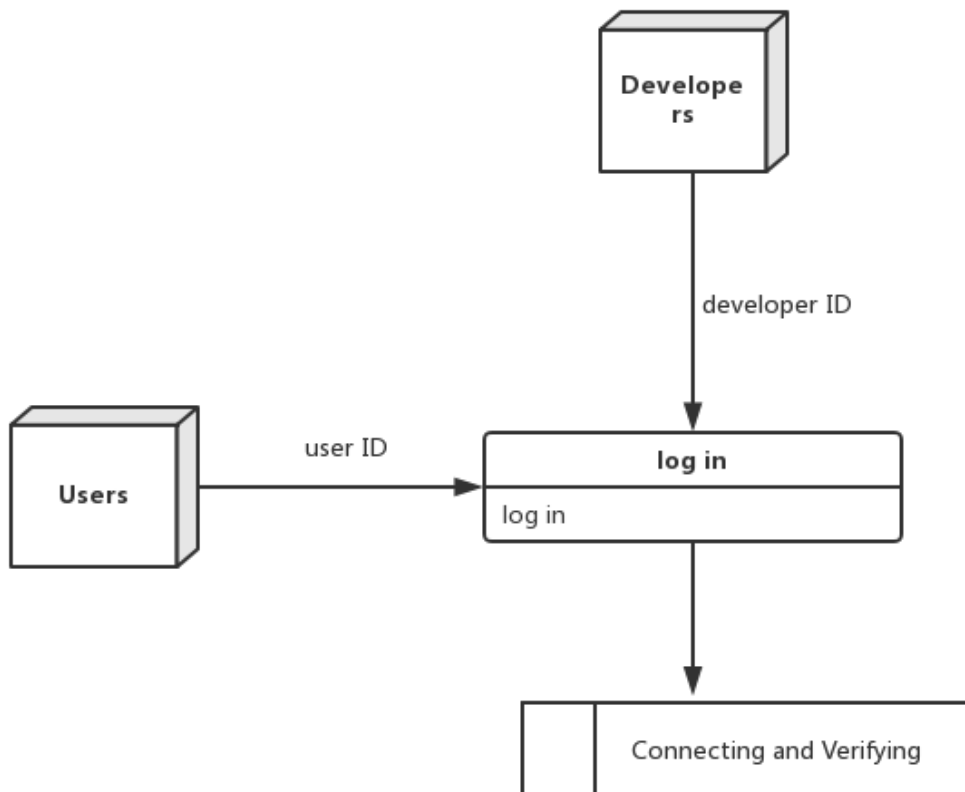
System Context Data Flow Diagram (Full-Bodied)



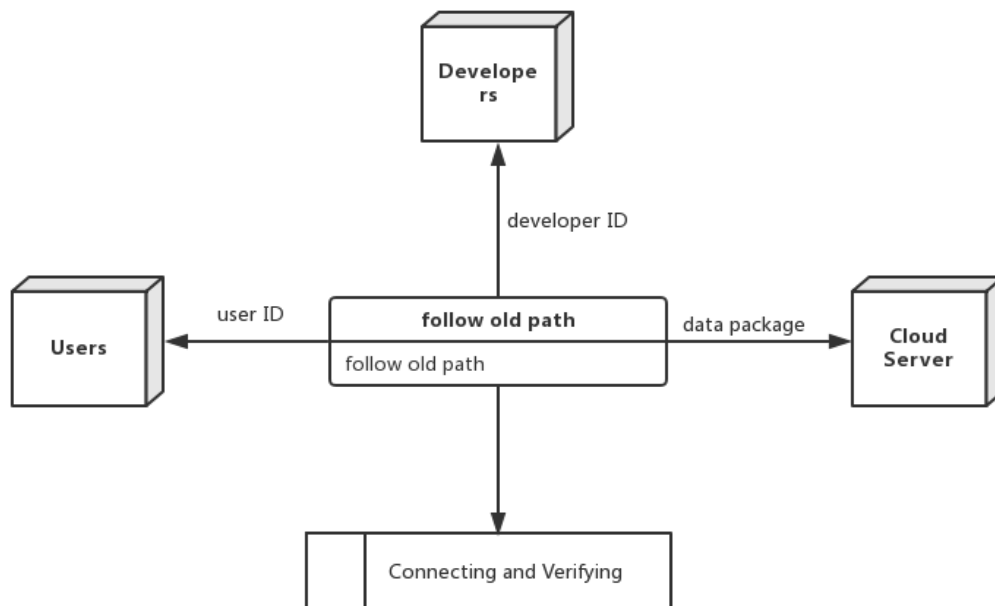
Event Diagram (Creating New Path)



Event Diagram (Logging In)



Event Diagram (Following Old Path)



The event diagram shows only the event handler and the inputs and outputs for each other. One or more system diagrams are constructed by merging the event diagram. We can intuitively observe the relationship between events from the graph.

System Diagram

